

Computational Methods in Applied Sciences

Krzysztof Arczewski  
Wojciech Blajer  
Janusz Frączek  
Marek Wojtyra *Editors*

# Multibody Dynamics

Computational Methods and  
Applications



# Multibody Dynamics

# Computational Methods in Applied Sciences

---

Volume 23

---

## *Series Editor*

E. Oñate

International Center for Numerical Methods in Engineering (CIMNE)

Technical University of Catalunya (UPC)

Edificio C-1, Campus Norte UPC

Gran Capitán, s/n

08034 Barcelona, Spain

onate@cimne.upc.edu

www.cimne.com

Krzysztof Arczewski • Wojciech Blajer  
Janusz Frączek • Marek Wojtyra  
Editors

# Multibody Dynamics

Computational Methods and Applications



*Editors*

Krzysztof Arczewski  
Warsaw University of Technology  
Faculty of Power and Aeronautical  
Engineering  
ul. Nowowiejska 24  
00-665 Warsaw  
Poland  
krisarcz@meil.pw.edu.pl

Janusz Frączek  
Warsaw University of Technology  
Faculty of Power and Aeronautical  
Engineering  
ul. Nowowiejska 24  
00-665 Warsaw  
Poland  
jfraczek@meil.pw.edu.pl

Wojciech Blajer  
Technical University of Radom  
Faculty of Mechanical Engineering  
ul. Krasickiego 54  
26-600 Radom  
Poland  
w.blajer@pr.radom.pl

Marek Wojtyra  
Warsaw University of Technology  
Faculty of Power and Aeronautical  
Engineering  
ul. Nowowiejska 24  
00-665 Warsaw  
Poland  
mwojtyra@meil.pw.edu.pl

ISSN 1871-3033

ISBN 978-90-481-9970-9

e-ISBN 978-90-481-9971-6

DOI 10.1007/978-90-481-9971-6

Springer Dordrecht Heidelberg London New York

Library of Congress Control Number: PCN applied for

© Springer Science+Business Media B.V. 2011

No part of this work may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, microfilming, recording or otherwise, without written permission from the Publisher, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work.

Printed on acid-free paper

Springer is part of Springer Science+Business Media ([www.springer.com](http://www.springer.com))

# Preface

Multibody Dynamics is an exciting area of applied and computational mechanics, whose substantial progress during the last five decades has stemmed from the rapid and simultaneous development of many technological disciplines like robotics, spacecraft and machine design, and was stimulated by the advances in computational techniques. In order to deliver methods and tools for the modeling, analysis and simulation of complex mechanical systems, various topics were merged in the field, including contact and impact, control and mechatronics, real-time simulation, optimization, flexibility, time integration schemes and software development. The current area of interest include robotics and walking machines, road and railway vehicle dynamics, aerospace, biomechanics, and many other multidisciplinary applications.

The ECCOMAS Thematic Conference on Multibody Dynamics was initiated in Lisbon in 2003, and then continued in Madrid (2005) and Milan (2007), aimed at providing a venue for exchanging ideas and recent developments related to the theory and applications of multibody systems. The fourth edition of the Conference was held at the Warsaw University of Technology, Warsaw, Poland, from June 29 to July 2, 2009. At the Conference participated 219 researches from 27 countries, mainly from Europe (162), but also from Asia (40), and North (13) and South America (4). They presented 167 technical papers, having an excellent forum for discussion and technical exchange on the most recent advances in the rapidly growing field of Multibody Dynamics.

The present book is a collection of revised and extended versions of 15 papers presented at the Conference, recommended by the Session Organizers for publication in this post-conference book. The general selection criterion was that the papers best reflect the state-of-art of the topics associated to the particular sessions, and cover the areas of biomechanics (Raison et al.), contact dynamics (Flores et al./Ziegler and Eberhard), control, mechatronics and robotics (Iwamura et al./Seifried), efficient methods and real-time applications (Cavagna et al./Pfau and Schaden), flexible multibody dynamics (Ambrósio et al./Dibold and Gerstmayr), formulations and numerical methods (García Orden and Aguilera/Schindler et al.), miscellaneous multibody applications (Frączek and Wojtyra), optimization

(Brüls et al.), software development, validation, and education (Tasora et al.), and vehicle systems (Bottasso et al.). We hope you will find the reading of this collection enjoyable and stimulating.

March 2010  
Radom/Warsaw

Wojciech Blajer, Krzysztof Arczewski  
Janusz Frączek, and Marek Wojtyra

# Contents

<b>A Flexible Multibody Pantograph Model for the Analysis of the Catenary–Pantograph Contact</b> .....	1
Jorge Ambrósio, Frederico Rauter, João Pombo, and Manuel S. Pereira	
<b>Maneuvering Multibody Dynamics: New Developments for Models with Fast Solution Scales and Pilot-in-the-Loop Effects</b> .....	29
Carlo L. Bottasso, Giorgio Maisano, and Francesco Scorcelletti	
<b>Optimization of Multibody Systems and Their Structural Components</b> .....	49
Olivier Brûls, Etienne Lemaire, Pierre Duysinx, and Peter Eberhard	
<b>Real-Time Aeroservoelastic Analysis of Wind-Turbines by Free Multibody Software</b> .....	69
Luca Cavagna, Alessandro Fumagalli, Pierangelo Masarati, Marco Morandini, and Paolo Mantegazza	
<b>Comparison of Planar Structural Elements for Multibody Systems with Large Deformations</b> .....	87
Markus Dibold and Johannes Gerstmayr	
<b>Modeling and Analysis of Rigid Multibody Systems with Translational Clearance Joints Based on the Nonsmooth Dynamics Approach</b> .....	107
Paulo Flores, Remco Leine, and Christoph Glocker	
<b>Application of General Multibody Methods to Robotics</b> .....	131
Janusz Frączek and Marek Wojtyra	



<b>Energy Considerations for the Stabilization of Constrained Mechanical Systems with Velocity Projection</b> .....	153
Juan C. García Orden and Roberto A. Ortega Aguilera	
<b>A General Purpose Algorithm for Optimal Trajectory Planning of Closed Loop Multibody Systems</b> .....	173
Makoto Iwamura, Peter Eberhard, Werner Schiehlen, and Robert Seifried	
<b>Real-Time Simulation of Extended Vehicle Drivetrain Dynamics</b> .....	195
Ralf U. Pfau and Thomas Schaden	
<b>Assessment of Antagonistic Muscle Forces During Forearm Flexion/Extension</b> .....	215
Maxime Raison, Christine Detrembleur, Paul Fisette, and Jean-Claude Samin	
<b>Computing Time Reduction Possibilities in Multibody Dynamics</b> .....	239
Thorsten Schindler, Markus Friedrich, and Heinz Ulbrich	
<b>Optimization-Based Design of Minimum Phase Underactuated Multibody Systems</b> .....	261
Robert Seifried	
<b>GPU-Based Parallel Computing for the Simulation of Complex Multibody Systems with Unilateral and Bilateral Constraints: An Overview</b> .....	283
Alessandro Tasora, Dan Negrut, and Mihai Anitescu	
<b>Investigation of Gears Using an Elastic Multibody Model with Contact</b> .....	309
Pascal Ziegler and Peter Eberhard	

# A Flexible Multibody Pantograph Model for the Analysis of the Catenary–Pantograph Contact

Jorge Ambrósio, Frederico Rauter, João Pombo, and Manuel S. Pereira

**Abstract** The pantograph–catenary system is still the most reliable form of collecting electric energy for running trains. This system should ideally run with relatively low contact forces, in order to minimize wear and damage of the contacting elements but without contact loss to avoid power supply interruption and electric arching. However, the quality of the pantograph–catenary contact may be affected by operational conditions, defects on the overhead equipment, environmental conditions or by the flexibility of the pantograph components. In this work a flexible multibody methodology based on the use of the mean-axis conditions, as reference conditions, mode component synthesis, as a form of reducing the number of generalized coordinates of the system and virtual bodies, as a methodology to allow the use of all kinematic joints available for multibody modeling and application of external forces, are used to allow building the flexible multibody pantograph models. The catenary model is built in a linear finite element code developed in a Matlab environment, which is co-simulated with the multibody code to represent the complete system interaction. A thorough description of rigid-flexible multibody pantograph models is presented in a way that the proposed methodology can be used. Several flexible multibody models of the pantograph are described and proposed and the quality of the pantograph–catenary contact is analyzed and discussed in face of the flexibility of the overhead components.

## 1 Introduction

The interaction between the pantograph and the catenary is one of the factors that limits the operating speed of railway vehicles and, consequently, is one of the research priorities in the European railway community. These limitations

---

J. Ambrósio (✉), F. Rauter, J. Pombo, and M.S. Pereira  
IDMEC-IST, Av. Rovisco Pais 1, Lisboa, Portugal  
e-mail: [jorge@dem.ist.utl.pt](mailto:jorge@dem.ist.utl.pt); [frauter@dem.ist.utl.pt](mailto:frauter@dem.ist.utl.pt); [jpombo@dem.ist.utl.pt](mailto:jpombo@dem.ist.utl.pt);  
[mpereira@dem.ist.utl.pt](mailto:mpereira@dem.ist.utl.pt)

concern not only the ability to collect energy at high operating speeds but also the interoperability between the overhead equipment in trains and infrastructure. From the mechanical point of view, the most important feature of the pantograph–catenary system consists on the contact quality between the registration strips of the pantograph and the contact wires of the catenary. The system must ideally run with relatively low contact forces, to minimize wear and damage of the contact elements, but with high enough contact forces to prevent contact loss, to ensure a constant power supply and minimize the occurrence of electric arching. The design of pantograph–catenary systems aims at controlling the interaction phenomena maintaining the contact forces within an acceptable operational envelope. Among the factors that affect the quality of the pantograph–catenary contact are those concerned with the defects on the catenary or pantograph, environmental conditions, such as wind [1, 2] and extreme temperatures [3], running dynamics of the railway vehicle and the deformability of the pantograph mechanical system. This work proposes the use of flexible multibody methodologies to describe the pantograph system and the co-simulation of the models obtained with detailed finite element models of the catenary to evaluate the quality of the overhead contact and to identify the main mechanical issues influencing it.

Some of the earlier works in flexible multibody systems use fixed reference frames to describe the small elastic deformations given by the finite element method of planar mechanical systems [4]. This methodology effectively coupled the rigid body motion and the small deformations. To be able to analyze complex shaped flexible multibody systems, Shabana and Wehage proposed the use of substructuring and the model component synthesis method to reduce the number of generalized coordinates required to represent the flexible components [5, 6]. Reduction of the dimension of the flexible multibody problem is achieved by choosing only a small number of suitable vibration modes. In most cases only a small number of natural modes of vibration are needed, namely those related to the lower natural frequencies of the structure. The static correction modes represent the typical response of a structure subject to given boundary conditions [7]. Criteria to estimate the number of modes of vibration of each type have been proposed and proven to be successful for low velocity systems [8].

The coupling terms is dependent of the type of finite elements used in the model and involve the derivation of the element shape functions, which are not available in finite element literature [9, 10]. To enable the general use of finite element types in the analysis of flexible multibody systems a lumped mass formulation, based on the diagonalization of the mass matrix preserving the rotational inertia, is used [11].

For problems in which the flexible bodies experience nonlinear deformation the approach must be different and based on large displacements and rotations theory [12–14]. For nonlinear problems Kane and coworkers propose a nonlinear theory that includes the dynamic stiffening [15]. In the line of work developed by the finite element community [16–18], Cardona and Geradin propose a formulation for the nonlinear flexible bodies using either exact geometrical models or substructuring [19, 20]. An approach based on the use of the finite rotations nodal coordinates enabling the capture of the geometric nonlinear deformations has been proposed

and named absolute nodal coordinates [21]. A different approach has been proposed to model geometric nonlinear deformations based on relaxing the need to exhibit small moderate rotations about the reference frame, by using an incremental finite-element approach within the flexible body description [22]. The extent of the use of the referred approach to model material nonlinearities has also been proposed [23]. In these formulations the rigid body motion and the elastic variables are expressed in inertia reference frames, where the deformation state is derived with respect to local reference frames and a relation deformation-displacement is obtained but, due to the inherent nonlinearity of the deformations, the problem cannot be reduced implying the handling of large system matrices during the analysis process. Because the deformations observed in the pantograph or on the catenary are small, only linear elastodynamics is considered in the flexible multibody models used. In any case, structural damping is used in order to improve the time integration [24].

The use of finite element method on the framework of flexible multibody dynamics implies the definition of a set of reference conditions. Straightforward reference conditions are the body fixed reference frames, where the frame is attached to one or more nodes of the flexible body, constraining at least six degrees of freedom [11]. The mean-axis reference conditions correspond to a different approach by introducing a floating frame defined to minimize the kinetic energy associated to the deformation, measured with respect to an observer stationed on the flexible body [25, 26]. Another type of floating reference frame is called the principal axis reference conditions where the origin of the reference frame is associated to the instantaneous center of mass and its directions to the principal inertia directions [27]. Augusta and Ambrósio analyze and point out the major advantages and drawbacks of the different methodologies and their main applications [28]. The mean-axis conditions are used here as reference conditions for the flexible multibody formulation.

To be able to use the extensive library of kinematic constraints developed for rigid multibody systems with flexible bodies, the concept of virtual bodies has been developed [29, 30]. The numerical efficiency of this methodology applied to complex structures was shown using a sparse matrix solver after comparison to that of multibody models with custom developed flexible kinematic joints [31]. Kinematic joints based on the use of the virtual body approach [32] are implemented in this work so that the pantograph model can include the type of kinematic joints particular to its construction, described in reference [32].

Previous studies of the catenary–pantograph interaction emphasized not only the mechanical aspects of construction, operation and maintenance but also the challenges of its numerical simulation due to the multi-physics characteristic of this problem. Ranging from simple linear catenary models using 2D finite elements [33], or having the catenary represented by cables and loaded by a lumped mass model of a pantograph [34] several approaches proposed kept the problem simple enough to tackle it by a single code. In a similar line of work Dahlberg describes the contact wire as an axially loaded beam and uses modal analysis to represent its deflection when subjected to transversal and axial loads, showing in the process its relation to the critical velocity of the pantograph [35]. Labergri presents a very thorough description of the pantograph–catenary system that includes a 2D

finite element model for the catenary and a multibody pantograph, being the contact treated by unilateral constraints [36]. In all works mentioned it is claimed that the catenary structural deformations are basically linear and consequently the catenary systems are modeled using linear finite elements. The slacking of droppers is an exception being handled as a nonlinear effect. Another approach is proposed by Seo and coworkers that states the need to treat the catenaries as being nonlinear due to their large deformations [37, 38]. The large deformation of the catenary is modeled using the three-dimensional finite element absolute nodal coordinate formulation while the pantograph is a full 3D multibody model. The interaction between the pantograph and the catenary is modeled by a sliding joint that allows for the motion of the pan-head on the catenary cable and no contact loss is represented [38]. Arnold and Simeon address the pantograph–catenary interaction as time dependent problems coupled by constraints on boundaries [39]. A half-explicit integration method reversible in time was also developed in order to preserve as much information as possible during time discretization. Due to the multi-physics problem involved in the catenary–pantograph system, Veitl and Arnold proposed a co-simulation strategy between the code PROSA, where a catenary is described by the finite difference method and the SIMPACK commercial multibody code, used to simulate the pantograph [40]. Several strategies tackling the co-simulation problem, such as gluing algorithms proposed by Hulbert and coworkers [41] or the co-simulation procedures suggested by Kubler and Schiehlen [42]. Recognizing that the finite element method is appropriate to model in detail the catenary and that the multibody dynamics approach is suited to handle the pantograph dynamics, a co-simulation approach using two separate codes is proposed in this work.

## 2 Flexible Multibody Systems

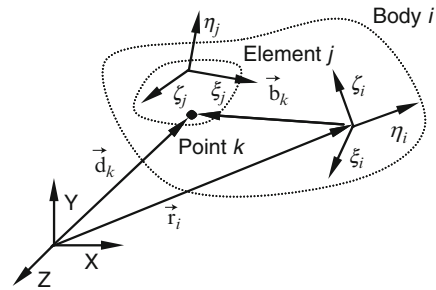
### 2.1 Flexible Body Equations of Motion

For the flexible body depicted by Fig. 1 let  $\mathbf{q}_i = [\mathbf{q}_r^T \mathbf{u}'^T]^T$  be the vector of generalized coordinates of body  $i$ , where  $\mathbf{q}_r = [\mathbf{r}_i^T \mathbf{p}_i^T]^T$  represents the translational and rotational position of body  $i$  local coordinate system  $(\xi, \eta, \zeta)_i$  and vector  $\mathbf{u}'$  represents body  $i$  elastic coordinates. The flexible body equations of motion are obtained by Gonçalves and Ambrósio as [11].

$$\begin{bmatrix} \mathbf{M}_{rr} & \mathbf{M}_{r\phi} & \mathbf{M}_{rf} \\ \mathbf{M}_{\phi r} & \mathbf{M}_{\phi\phi} & \mathbf{M}_{\phi f} \\ \mathbf{M}_{fr} & \mathbf{M}_{f\phi} & \mathbf{M}_{ff} \end{bmatrix}_i \begin{bmatrix} \ddot{\mathbf{r}}_i \\ \dot{\omega}'_i \\ \ddot{\mathbf{u}}'_i \end{bmatrix} = \begin{bmatrix} \mathbf{g}_r \\ \mathbf{g}'_\phi \\ \mathbf{g}'_f \end{bmatrix}_i + \begin{bmatrix} \mathbf{s}_r \\ \mathbf{s}'_\phi \\ \mathbf{s}'_f \end{bmatrix}_i - \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K}_i \end{bmatrix} \begin{bmatrix} \mathbf{r}_i \\ \mathbf{p}_i \\ \mathbf{u}'_i \end{bmatrix} \quad (1)$$

where the mass matrix  $\mathbf{M}_i$  contains the mass, inertia tensor and inertia coupling terms, vector  $\mathbf{s}_i$  represents the velocity quadratic terms and other acceleration independent terms,  $\mathbf{g}_i$  is the generalized external force vector, and  $\mathbf{K}_i$  is the finite

**Fig. 1** Flexible body *i*



elements stiffness matrix. The mass matrix in Eq.(1) may be either consistent or lumped. In order to maintain the inertia coupling terms independent of the finite element shape functions, the lumped mass formulation is used in this work [11].

The equations of motion obtained, using consistent or diagonalized mass matrices, do not have a unique displacement field. It is necessary to impose a set of reference conditions to eliminate the rigid body modes and provide the unique displacement field of the flexible body. In general, reference conditions are written as kinematic constraints that relate the independent and the dependent elastic coordinates. The mean axis conditions constraints are such that enforce the local frame  $(\xi, \eta, \zeta)_i$  of body  $i$  to follow the motion of the nodes in such a way that the kinetic energy associated with the deformation corresponds to a minimum value for an observer stationary in the body local frame [25, 26]. The deformation kinetic energy of a flexible body can be expressed in terms of the generalized elastic coordinates with respect to the local coordinate system  $(\xi, \eta, \zeta)_i$  as:

$$T = \frac{1}{2} \dot{\mathbf{u}}'^T \mathbf{M} \dot{\mathbf{u}}' = \frac{1}{2} \sum_{k=1}^n m_k \dot{\delta}_k'^T \dot{\delta}_k' + \frac{1}{2} \sum_{k=1}^n \dot{\theta}_k'^T \mu_k \dot{\theta}_k' \quad (2)$$

where the nodal translation velocities are denoted by  $\dot{\mathbf{\delta}}'_k$  and the nodal angular velocities by  $\dot{\boldsymbol{\theta}}'_k$ . The generalized elastic coordinate velocities  $\dot{\mathbf{u}}'_k$  of a node  $k$  of the body mesh are written in terms of generalized set of coordinates  $\mathbf{q}_r$  of the body as:

$$\dot{\mathbf{u}}'_k = \begin{bmatrix} \dot{\delta}'_k \\ \dot{\theta}'_k \end{bmatrix} = \begin{bmatrix} \mathbf{A}^T (\dot{\mathbf{d}}_k - \dot{\mathbf{r}} + \mathbf{A} \tilde{\mathbf{b}}'_k \omega') \\ \dot{\alpha}'_k - \omega' \end{bmatrix} \quad (3)$$

in which matrix  $\mathbf{A}$  represents the transformation matrix from the body local coordinate system to the inertial frame. Minimizing the deformation kinetic energy of the body with respect to the translational and rotational velocities leads to

$$T_{\dot{\mathbf{r}}} = \frac{\partial T}{\partial \dot{\mathbf{r}}} = \sum_{k=1}^n m_k (\dot{\mathbf{d}}_k - \dot{\mathbf{r}} + \mathbf{A} \tilde{\mathbf{b}}'_k \boldsymbol{\omega}') = 0$$

$$T_{\omega'} = \frac{\partial T}{\partial \omega'} = \sum_{k=1}^n m_k (\mathbf{A} \tilde{\mathbf{b}}'_k)^T (\dot{\mathbf{d}}_k - \dot{\mathbf{r}} + \mathbf{A} \tilde{\mathbf{b}}'_k \omega') + \sum_{k=1}^n \mu_k (\dot{\alpha}'_k - \omega') = 0 \quad (4)$$

Substituting Eq. (3) into Eq. (4) results in the velocity constraint equations that define the mean axis reference conditions, as

$$\dot{\Phi}^{(ma)} \equiv \left\{ \begin{array}{c} \sum_{k=1}^n m_k \dot{\delta}'_k \\ \sum_{k=1}^n m_k \tilde{\mathbf{b}}_k'^T \dot{\delta}'_k + \sum_{k=1}^n \mu_k \dot{\theta}'_k \end{array} \right\} = \mathbf{0} \quad (5)$$

The velocity constraint equations may be written in more compact form as:

$$\dot{\Phi}^{(ma)} \equiv \Phi_{\mathbf{u}'}^{(ma)} \dot{\mathbf{u}}' = \mathbf{0} \quad (6)$$

where  $\Phi_{\mathbf{u}'}^{(ma)}$  represents the Jacobian matrix of the mean axis reference conditions constraint equations, which is explicitly written as

$$\Phi_{\mathbf{u}'}^{(ma)} = \begin{bmatrix} \Phi_{\delta'}^{(ma)} \\ \Phi_{\theta'}^{(ma)} \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^n m_k \mathbf{I}_k^T & \mathbf{0}^T \\ \sum_{k=1}^n m_k \tilde{\mathbf{b}}_k' \mathbf{I}_k^T & \sum_{k=1}^n \mu_k \mathbf{I}_k^T \end{bmatrix} \quad (7)$$

The time derivative of Eq. (6) results in the acceleration constraint equations of the mean axis reference conditions, written here in a compact form as

$$\ddot{\Phi}^{(ma)} \equiv \Phi_{\mathbf{u}'}^{(ma)} \ddot{\mathbf{u}}' = \gamma^{(ma)} \quad (8)$$

The constraints associated to the mean axis conditions are imposed on the flexible body equations of motion, described by Eq. (1), leading to

$$\begin{bmatrix} \mathbf{M}_r & \mathbf{0} & \mathbf{0} \\ \mathbf{M}_{fr} & \mathbf{M}_{ff} & \Phi_{\mathbf{u}'}^{(ma)T} \\ \mathbf{0} & \Phi_{\mathbf{u}'}^{(ma)} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_r \\ \ddot{\mathbf{u}}' \\ \lambda^{(ma)} \end{bmatrix} = \begin{bmatrix} \mathbf{g}_r \\ \mathbf{g}'_f \\ \gamma^{(ma)} \end{bmatrix} - \begin{bmatrix} \mathbf{s}_r \\ \mathbf{s}'_f \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \mathbf{K}_f \mathbf{u}' \\ \mathbf{0} \end{bmatrix} \quad (9)$$

Note that the mean axis conditions are non-holonomic constraint conditions and can only be defined at the velocity and acceleration levels.

The flexible body equations of motion, shown in Eq. (9), include a very large number of generalized coordinates, leading to a computational expensive procedure. For linear elastic small deformations, as those experienced by the pantograph components in the applications foreseen in this work, it is possible to represent the deformation of the flexible body as a sum of deformation modes that are constant in time. Let those deformation modes be the modes of vibration associated to the natural frequencies of the flexible body. The generalized elastic coordinates of body  $i$  are now described by a weighted sum of these modes as

$$\mathbf{u}' = \mathbf{X} \mathbf{w} \quad (10)$$

where  $\mathbf{w}$  represents the contributions of the modes of vibration towards the nodal displacements and  $\mathbf{X}$  the modal matrix containing a selected number of modes of vibration  $\chi_i$  that are obtained by solving the eigenproblem:

$$\mathbf{K}_{ff}\chi_i = \bar{\omega}_i \mathbf{M}_{ff}\chi_i \quad (11)$$

The solution of Eq. (11) is independent of the reference conditions used to constrain the rigid body movement of the elastic coordinates. Therefore the modes of vibration obtained correspond to those of a structure free in space, defined as free-free modes. The vibration modes obtained related to the first six lowest frequencies, generally null, represent the rigid body motion of the flexible body. These modes of vibration are removed from the modal matrix. A simpler system of equations is obtained by normalizing  $\chi_i$  with respect to the mass matrix  $\mathbf{M}_{ff}$

$$\mathbf{X}^T \mathbf{M}_{ff} \mathbf{X} = \mathbf{I} \quad (12)$$

$$\mathbf{X}^T \mathbf{K}_{ff} \mathbf{X} = \mathbf{\Lambda} \quad (13)$$

where  $\mathbf{\Lambda}$  is a diagonal matrix containing the square the natural frequencies associated to each mode of vibration.

By substituting Eq. (10), and its time derivatives, into Eq. (9), pre-multiplying the second row by  $\mathbf{X}^T$  and using the relations described by Eqs. (12) and (13) leads to:

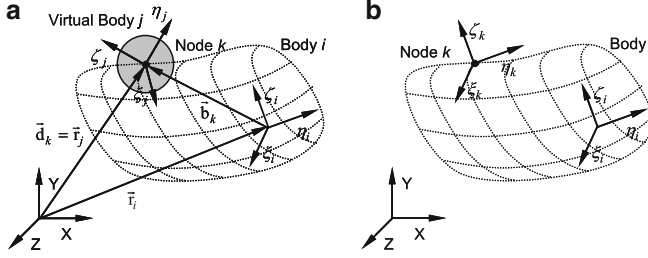
$$\begin{bmatrix} \mathbf{M}_r & \mathbf{M}_{rf}\mathbf{X} & \Phi_{q_r}^T \\ \mathbf{X}^T \mathbf{M}_{fr} & \mathbf{I} & \mathbf{X}^T \Phi_{q_f}^T \\ \Phi_{q_r} & \Phi_{q_f} \mathbf{X} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_r \\ \ddot{\mathbf{w}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{g}_r \\ \mathbf{X}^T \mathbf{g}'_f \\ \gamma \end{bmatrix} - \begin{bmatrix} \mathbf{s}_r \\ \mathbf{X}^T \mathbf{s}'_f \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \mathbf{\Lambda} \mathbf{w} \\ \mathbf{0} \end{bmatrix} \quad (14)$$

The number of generalized elastic coordinates used in Eq. (14) is equal to the number of vibration modes included in the modal matrix, thus being possible to reduce considerably the problem dimension considering a general use of flexible multibody models. The effects of local deformations induced by high concentrated loads originated, for example, by kinematic constraint reaction forces or other force elements, can also be included in the modal synthesis using of static correction modes [7, 8].

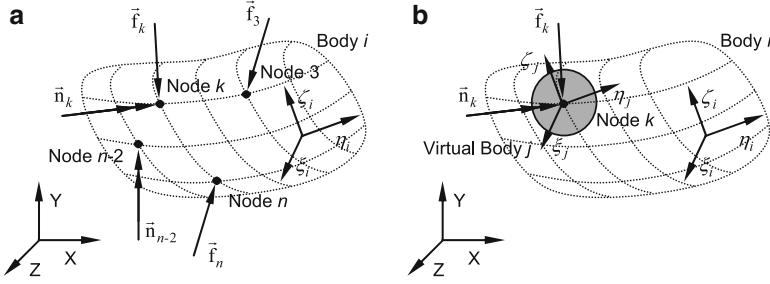
## 2.2 Kinematic Joints with Virtual Bodies

The use of flexible bodies requires that the kinematic joints implemented in the multibody code are re-written again for the new set of generalized coordinates used. A form to circumvent this difficulty is to use the concept of virtual bodies introduced by Bae et al. [30] and further developed by Ambrosio and Gonçalves [31, 32]. With the virtual body approach, a rigid joint between a flexible body and a rigid body is derived for a node  $k$  of the mesh of the flexible body and the origin of the virtual rigid body fixed reference frame, as depicted in Figs. 2 and 3. Afterwards, any





**Fig. 2** Rigid joint between a flexible body  $i$  and a massless rigid body  $j$ : (a) virtual body; (b) nodal and body fixed coordinate systems



**Fig. 3** External forces applied on a flexible body: (a) Forces applied directly on the nodes; (b) FORCES applied on the virtual body attached to the finite element nodes

kinematic joint between the flexible body and other bodies of the system is established using the virtual body instead, making possible to use any of the kinematic joints available on the multibody code library.

The constraint equations for the rigid joint are defined for the translational and rotational parts of the constraint independently. Let a spherical joint be defined between node  $k$  of flexible body  $i$  and a point  $P$  of the rigid body  $j$ , coincident to the origin of the body fixed coordinate system. This is the translation part of the rigid kinematic joint written as

$$\Phi^{(t)} \equiv \mathbf{r}_j - \mathbf{r}_i - \mathbf{A}_i \mathbf{b}_k = \mathbf{0} \quad (15)$$

In order to define the rotational part of the rigid joint let a coordinate system  $(\xi, \eta, \zeta)_k$  be attached to node  $k$ , as showed in Fig. 3b. The nodal frame is defined by unit vectors  $\bar{\mathbf{e}}_k = [\bar{\mathbf{e}}_k^1 \ \bar{\mathbf{e}}_k^2 \ \bar{\mathbf{e}}_k^3] \equiv \mathbf{I}$  initially parallel to the flexible body  $i$  local reference frame  $(\xi, \eta, \zeta)_i$  unit vectors  $\mathbf{e}'_i$ . Unit vectors defining both nodal and body coordinate frames are expressed in the inertial frame  $(X, Y, Z)$  as:

$$\mathbf{e}_k = \mathbf{A}_i \mathbf{A}_k \bar{\mathbf{e}}_k \quad (16)$$

$$\mathbf{e}_j = \mathbf{A}_j \mathbf{e}'_j \quad (17)$$

where  $\mathbf{A}_k = \mathbf{I} + \tilde{\theta}'_k$  represents the nodal rotational matrix for small rotations. The rotational part of the rigid joint constraint enforces that the relative orientation between the node and virtual body reference frames remains invariant, i.e.

$$\Phi^{(r)} \equiv (\mathbf{A}_i \mathbf{A}_k \tilde{\mathbf{e}}_k^m)^T \mathbf{A}_j \tilde{\mathbf{e}}_j^l - \beta^{ml} = \mathbf{0}; \text{ with } (m, l) = (2, 1), (3, 1), (2, 3) \quad (18)$$

in which only three equations are defined, corresponding to the independent rotational constraints. Constants  $\beta^{ml}$  are related to the initial angle between the axis of the coordinate systems in the undeformed state.

The external applied forces on the flexible bodies are applied to the nodes of the finite element model, as shown in Fig. 3a. Assume that force  $\mathbf{f}_i$  and a moment  $\mathbf{n}_i$ , shown in Fig. 3a, are applied in node  $k$  of the flexible body. Then, introducing a virtual body rigidly attached to that node allows for the direct applications of these forces on the center of mass, as shown in Fig. 3b. Note that the use of the virtual body approach allows for setting rigid joints with more than one node at a time. This approach can be used to setup complex interaction conditions between the flexible body and the external environment.

### 3 Co-simulation of Multibody and Finite Element Codes

The fundamental element of the co-simulation between a finite element code, denominated by EUROPACAS-FE, and the multibody code, herein denominated by EUROPACAS-MB, is the contact module between the two subsystems. The contact force due to pantograph–catenary interaction is characterized by a high-frequency oscillating force with high relative amplitude. Railway industry measurement data shows that reasonable values for the contact force are, for a train running at approximately 80 m/s: a mean value of 200 N oscillating between 400 and 100 N. Loss of contact in particular points of the catenary may also occur. Therefore impact effects must be included in the model.

Most continuous force contact force models have similar features, i.e., they evaluate the contact force as a function of a pseudo-penetration between two elements and a proportionality factor often designated as stiffness of the contact elements. The contact model used here, proposed by Lankarani and Nikravesh [43], is of the Hertzian type and includes internal damping and relates the normal contact force  $f_n$  with the penetration between two rigid bodies  $\delta$  by

$$f_n = K \delta^n \left[ 1 + \frac{3(1-e^2)}{4} \frac{\dot{\delta}}{\dot{\delta}^{(-)}} \right] \quad (19)$$

where  $K$  is the generalized stiffness,  $e$  is the restitution coefficient,  $\dot{\delta}$  is the relative penetration velocity and  $\dot{\delta}^{(-)}$  is the relative impact velocity. Factor  $K$  is obtained from the Hertz contact theory as the external contact between two cylinders with perpendicular axis.

The issue of the co-simulation is now ‘reduced’ to be able to find the state variables of the finite element catenary model and of the multibody pantograph model at the same instants of time, so that the contact force and its application points can be evaluated and used in the equations of motion of each subsystem.

### 3.1 Integration of the Finite Elements Equations of Motion

The motion of the catenary is characterized by small rotations and small deformations, in which the only nonlinear effect is the slacking of the droppers, being typically modeled with linear finite elements. All catenary elements, contact and messenger wires are modeled by using Euler–Bernoulli beams. Using the finite element method, the equilibrium equations for the structural system are [44]

$$\mathbf{M} \mathbf{a} + \mathbf{C} \mathbf{v} + \mathbf{K} \mathbf{x} = \mathbf{f} \quad (20)$$

where  $\mathbf{M}$ ,  $\mathbf{C}$  and  $\mathbf{K}$  are the finite element global mass, damping and stiffness matrices of the finite element model of the catenary, not to be confused with the finite element models of the flexible bodies used for the pantograph. The nodal displacements vector is  $\mathbf{x}$  while  $\mathbf{v}$  is the vector of nodal velocities,  $\mathbf{a}$  is the vector of nodal accelerations and  $\mathbf{f}$  is the vector with the applied forces. Equation (20) is solved for  $\mathbf{x}$  or for  $\mathbf{a}$  depending on the integration method used.

In this work the integration of the nodal accelerations uses a Newmark family integration algorithm [45]. The contact forces are evaluated for  $t + \Delta t$  based on the position and velocity predictions for the FE mesh and on the pantograph predicted position and velocity. The finite element mesh accelerations are calculated by

$$(\mathbf{M} + \gamma \Delta t \mathbf{C} + \beta \Delta t^2 \mathbf{K}) \mathbf{a}_{t+\Delta t} = \mathbf{f}_{t+\Delta t} - \mathbf{C} \tilde{\mathbf{v}}_{t+\Delta t} - \mathbf{K} \tilde{\mathbf{d}}_{t+\Delta t} \quad (21)$$

Predictions for new positions and velocities of the nodal coordinates of the linear finite element model of the catenary are found as

$$\tilde{\mathbf{d}}_{t+\Delta t} = \mathbf{d}_t + \Delta t \mathbf{v}_t + \frac{\Delta t^2}{2} (1 - 2\beta) \mathbf{a}_t \quad (22a)$$

$$\tilde{\mathbf{v}}_{t+\Delta t} = \mathbf{v}_t + \Delta t (1 - \gamma) \mathbf{a}_t. \quad (22b)$$

Then, with the acceleration  $\mathbf{a}_{t+\Delta t}$  the positions and velocities of the finite elements at time  $t + \Delta t$  are corrected by

$$\mathbf{d}_{t+\Delta t} = \tilde{\mathbf{d}}_{t+\Delta t} + \beta \Delta t^2 \mathbf{a}_{t+\Delta t} \quad (23a)$$

$$\mathbf{v}_{t+\Delta t} = \tilde{\mathbf{v}}_{t+\Delta t} + \gamma \Delta t \mathbf{a}_{t+\Delta t}. \quad (23b)$$

This procedure is repeated until convergence is reached for a given time step.

### 3.2 Integration of the Multibody Equations of Motion

The forward dynamic analysis of a multibody system requires that the position vector  $\mathbf{q}^0$  and the velocity vector  $\dot{\mathbf{q}}^0$  are given. The multibody equations of motion assembled and solved for the unknown accelerations, which are in turn integrated in time together with the velocities. Due to the long periods of analysis and to the structure of the equilibrium equations not only the stabilization of the integration must be insured but also the constraint violations must be eliminated. In this work, the Baumgarte constraint stabilization method is used to stabilize the multibody system equations of motion and the Coordinate Partition Method is used to correct the position and velocity constraint equations when the violations exceed a prescribed acceptable tolerance [46], as depicted in Fig. 4.

The pantograph–catenary system is characterized by an intermittence of the contact between the contact wire of the catenary and the registration strip of the pantograph. The numerical methods used for the dynamic simulation must be able to represent the loss and start of contact. This fact puts particular restrictions on the numerical integration algorithms for both pantograph and catenary with particular emphasis on the time step size selection. The multibody code used for the pantograph dynamics, considered here, uses as a Gear multi-step multi-order integration algorithm [47, 48].

### 3.3 Co-simulation Using Different Codes

The analysis of the pantograph–catenary interaction is done by two independent codes, the pantograph code, EUROPACAS-MB, which uses a multibody

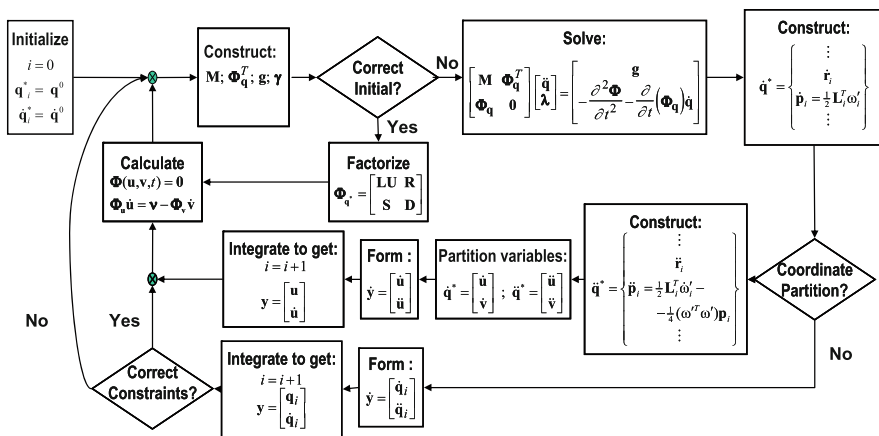
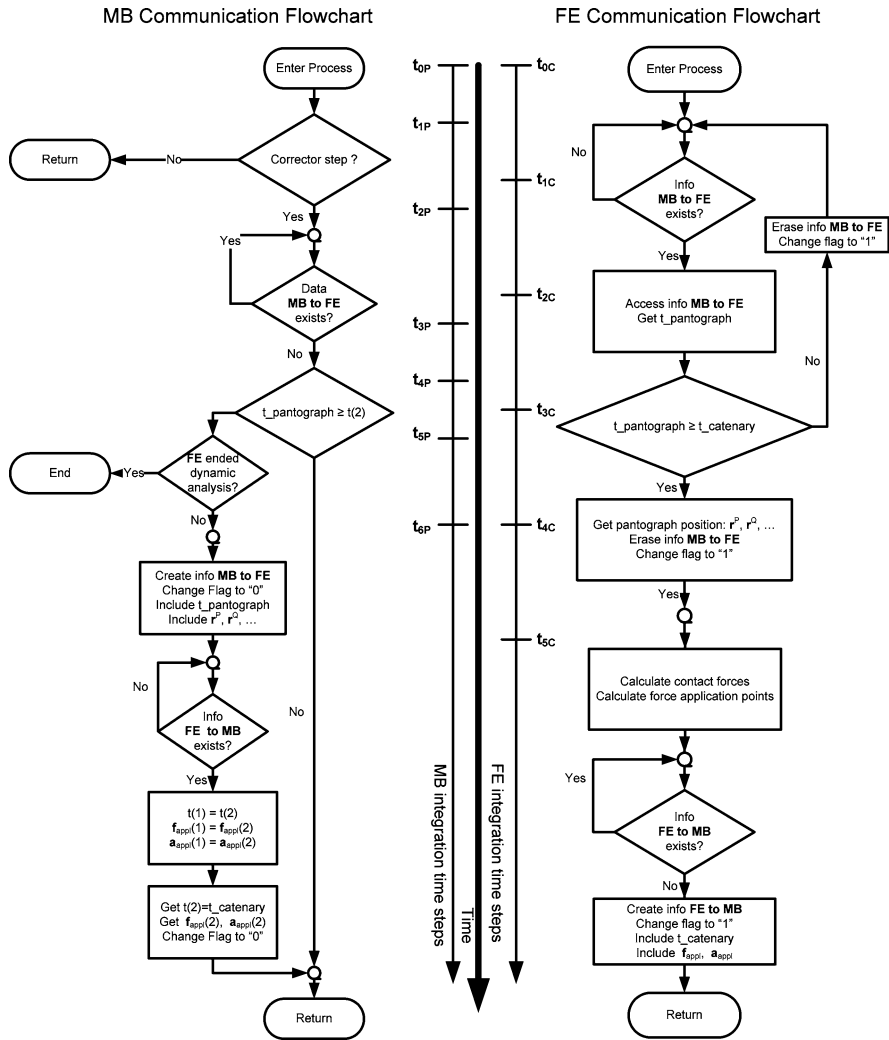


Fig. 4 Flowchart representing the forward dynamic analysis of a multibody system implemented





**Fig. 6** Communication procedure during the dynamic analysis

to be used in the FE analysis. Note that nowhere in the communication procedure outlined it is implied what kind of integration algorithm is used for the FE catenary analysis, provided that it is a fixed time step integrator. Even this condition can be relaxed, but it would not have any practical implication as it is not usual that FE dynamic analysis is performed with variable time step integrators.

4 Analysis of the Pantograph–Catenary Contact Problem

4.1 Pantograph Multibody Models

The flexibility of a pantograph is described by the experimental modes of vibration shown in Table 1 for the CX Pantograph [51]. The modal data acquisition is obtained by imposing a cyclical force to the pantograph head of constant value with frequencies ranging from 0 to 200 Hz. It is observed that important natural frequencies exist in the pantograph within the range of the operating frequencies of the overhead electric collection system, justifying that a flexible multibody approach is used to model the pantograph.

Several models of the pantograph, shown in Fig. 7, are modeled using a rigid-flexible multibody approach to evaluate their influence on the quality of the pantograph–catenary contact. The lower and top arms are steel tubular structures with varying cross-section, whereas the pantograph head is composed by steel, composite materials and carbon registration strips. Although highly detailed FEM models may be derived using solid and shell finite elements, simplified models of the referred bodies are used as local effect analysis as stress and strain analysis are not required. The mechanical data for the pantograph top arm and for its finite element model is shown in Table 2.

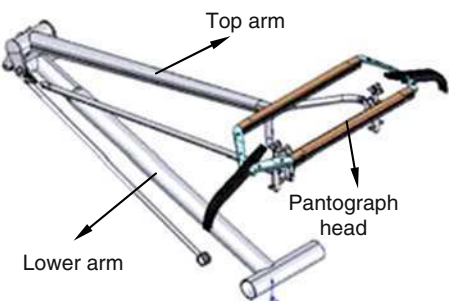
The modes of vibration of the top arm FEM model are obtained for free-free configuration, i.e., the model free in space. The first six structural natural frequencies are shown in Table 3.

The pantograph head, shown in Fig. 8, is another component for which the flexibility is expected to play a role. Its structure is composed by several elements with

Table 1 Representative experimental modal basis of the CX Pantograph

Mode Frequency			Mode Frequency		
n°	(Hz)	Description	n°	(Hz)	Description
1	11.0	Rotational movement of the main frame around base (Z)	2	19.3	Bending of the top link (Y)
3	40.1	Bending of the lower link (Y)	4	45.7	Bending of the panto-graph head (Z)
5	49.3	Bending of the top arm (Z)	6	71.1	Bending of the top link (Z)

**Fig. 7** CX pantograph used for the rigid-flexible multibody models



**Table 2** Material and geometric data for the upper arm and finite element information

Mechanical data	Unit	Value
Mass	[kg]	15.6
Young Modulus	[GPa]	206.8
Density	[kg/m <sup>3</sup> ]	7,820
Average length	[m]	1.36
FEM model data		Value
Finite element	[type]	3D elastic beam
Cross-section	[type]	Circular hollow
Cross-section area	[m <sup>2</sup> ]	$6.9 \times 10^{-4}$
Finite element		42
Nodes		43

different materials, including steel, for the support structures, carbon strips, for the registration strips, and composite materials, for the aerodynamic elements.

As the first mode of vibration of the pantograph head is a flexion mode, the structure may be modeled in a simple and straightforward way as a straight beam. The FEM model used is composed by a collection of beam elements, with rectangular cross-section and two lumped masses at the end-points of the straight beam, with the general characteristics shown in Table 4. The modes of vibration, for the free-free configuration, are described in Table 5.




In order to appraise the influence of the flexibility of the top arm to the global dynamic behavior of the pantograph in the rigid-flexible pantograph multibody model only the top arm is described as a flexible body. Figure 9 shows a representation of this multibody model, referred to as pantograph model 2.

By considering the top arm as a flexible body, four virtual bodies are added to the multibody model to allow for the definition of kinematic joints. In Table 6, the main characteristics of the flexible multibody model are described.

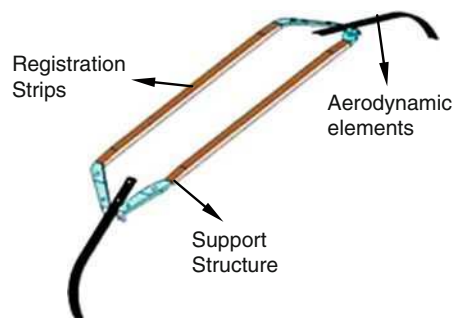
The characteristics of the rigid bodies used in the multibody model are presented in Table 7 for the fully rigid multibody model. Note that the top arm is flexible and, consequently, the mass and inertia is not explicitly given as input data. The virtual bodies shown in Table 8, and added to the rigid multibody model, are in the points of the top arm involving kinematic joints.



**Table 3** Natural frequencies and modes of vibration of the FEM model of the top arm

Mode n°	Frequency (Hz)	Description	n°	Mode (Hz)	Frequency	Description
1	89.9	First bending moment Z		2	100.8	First bending moment Y
3	219.9	Second bending moment Z		4	276.1	Second bending moment Y
5	358.5	Third bending moment Z		6	383.8	Forch opening

**Fig. 8** Pantograph head representation



**Table 4** Pantograph head FEM model characteristics

FEM model data	Unit	Value
Finite element		3D elastic beam
Cross-section		Rectangular
Cross-section area	[m <sup>2</sup> ]	$5 \times 10^{-4}$
Finite element		2
Finite element		Lumped mass
Finite element		2
Nodes		3

**Table 5** Natural frequencies and modes of vibration of the pantograph head FEM model

Mode n°	Frequency (Hz)	Description	Mode n°	Frequency (Hz)	Description
1	49.6	First bending moment X	2	98.7	Rotation X
3	173.9	Second bending moment X	4	335.0	First bending moment Z
5	398.7	Second rotation X	6	701.0	Translation + bending X

The virtual bodies, 8 through 11, are linked to the flexible bodies through the rigid kinematic joints and to the rigid bodies through standard kinematic joints. Table 9 presents the definition of the rigid kinematic joints for the rigid-flexible pantograph multibody model 2. Note that the rigid-flexible joints rigidly attach a node of the flexible body mesh to a virtual body. Consequently, the mesh of the flexible body must be generated in such a way that at the least one node is included at each point

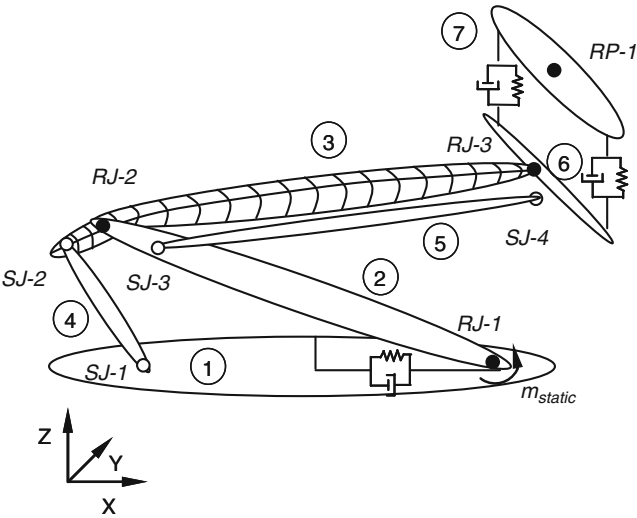


Fig. 9 Pantograph model 2 with a flexible top arm

Table 6 Characteristics of the pantograph multibody model 2

Multibody model data	Number
Rigid bodies	6
Virtual bodies	4
Flexible bodies	1
Rigid kinematic joints	8
Rigid-flexible kinematic joints	3

Table 7 Rigid body data of the pantograph multibody model 2

ID	Rigid body	Mass (kg)	Inertia properties (kg.m <sup>2</sup> )	Initial position (m)	Initial orientation
			$I_{\xi\xi}/I_{\eta\eta}/I_{\zeta\zeta}$	$x_0/y_0/z_0$	$e_1/e_2/e_3$
1	Pantograph base	32.65	2.76/4.87/2.31	0.00/0.00/0.00	0.00/0.00/0.00
2	Lower arm	32.18	0.31/10.43/10.65	-0.57/0.00/0.41	0.00/0.17/0.00
3	Upper arm	15.6	0.15/7.76/7.86	-0.39/0.00/1.06	0.00/-0.18/0.00
4	Lower link	3.10	0.05/0.46/0.46	-0.89/0.00/0.28	0.00/0.21/0.00
5	Upper link	1.15	0.05/0.48/0.48	-0.36/0.00/1.00	0.00/-0.16/0.00
6	Stab. arm	1.51	0.07/0.05/0.07	0.55/0.00/1.42	0.00/0.00/0.00
7	Panto. head	9.50	1.59/0.21/1.78	0.55/0.00/1.51	0.00/0.00/0.00

to which a kinematic joint is attached. The linear force elements are detailed in Table 10. The force exerted by the air pump, located between the lower arm and the base of the pantograph, is represented as a constant moment  $n_\eta = 440$  Nm applied to the lower arm.

**Table 8** Data for the virtual bodies used in the flexible pantograph multibody model 2

ID	Rigid body	Mass (kg)	Inertia properties (kg.m <sup>2</sup> )	Initial position (m)	Initial orientation
			$I_{\xi\xi}/I_{\eta\eta}/I_{\zeta\zeta}$	$x_0/y_0/z_0$	$e_1/e_2/e_3$
8	Virtual body	0.0	0.0/0.0/0.0	-1.19/0.00/-0.13	0.00/-0.18/0.00
9	Virtual body	0.0	0.0/0.0/0.0	1.01/-0.31/0.00	0.00/-0.18/0.00
10	Virtual body	0.0	0.0/0.0/0.0	-1.01/0.00/0.00	0.00/-0.18/0.00
11	Virtual body	0.0	0.0/0.0/0.0	1.01/0.31/0.00	0.00/-0.18/0.00

**Table 9** Definition of the kinematic joints used in flexible multibody model 2

ID	Kinematic joint	Connected bodies		Attachment points Local coordinates (m)	
		i	j	Body i ( $\xi_i/\eta_i/\zeta_i$ )	Body j ( $\xi_j/\eta_j/\zeta_j$ )
1	Revolute joint	1	2	(0.02/0.00/0.13) <sub>P</sub> (0.02/1.00/0.13) <sub>Q</sub>	(0.82/0.00/0.00) <sub>P</sub> (0.82/1.00/0.00) <sub>Q</sub>
2	Revolute joint	2	10	(-0.82/0.00/0.00) <sub>P</sub> (-0.82/1.00/0.00) <sub>Q</sub>	(0.00/0.00/0.00) <sub>P</sub> (0.00/1.00/0.00) <sub>Q</sub>
3	Revolute joint	11	6	(0.00/0.00/0.00) <sub>P</sub> (0.00/1.00/0.00) <sub>Q</sub>	(0.00/0.31/0.00) <sub>P</sub> (0.00/1.31/0.00) <sub>Q</sub>
4	Spherical joint	1	4	(-0.26/0.00/0.00) <sub>P</sub>	(0.69/0.00/0.00) <sub>P</sub>
5	Spherical joint	8	4	(0.00/0.00/0.00) <sub>P</sub>	(-0.62/0.00/-0.03) <sub>P</sub>
6	Spherical joint	2	5	(-0.78/0.00/0.00) <sub>P</sub>	(-1.00/0.00/0.00) <sub>P</sub>
7	Spherical joint	5	6	(0.96/0.00/0.00) <sub>P</sub>	(0.00/0.00/-1.05) <sub>P</sub>
8	Spherical joint	9	6	(0.00/0.00/0.00) <sub>P</sub>	(0.00/0.00/-1.05) <sub>P</sub>
9	Rev.-prism. joint	6	7	(0.00/0.34/0.00) <sub>P</sub> (0.00/0.34/1.00) <sub>Q</sub>	(0.00/0.34/0.00) <sub>P</sub> (0.00/1.34/0.00) <sub>Q</sub>

**Table 10** Linear force elements data for the flexible multibody model 2

ID element	Linear force	Spring elements		Damping coefficient (N.s/m)	Bodies		Attach points local coord (m)	
		Stiffness (N/m)	Length (m)		i	j	$\xi_i/\eta_i/\zeta_i$	$\xi_j/\eta_j/\zeta_j$
1	Spring-damper	2,000.00	9.06	3,000.00	1	2	0.28/0.00/0.09	0.82/0.00/-0.05
2	Spring-damper	3,600.00	0.12	13.00	6	7	0.00/0.34/0.00	0.00/0.34/0.00
3	Spring-damper	3,600.00	0.12	13.00	6	7	0.00/-0.34/0.00	0.00/-0.34/0.00

Another rigid-flexible pantograph multibody model, depicted as model 4 in Fig. 10 and described in Table 11, is used here considering the flexibility of the pantograph head only. The objective of this model is to understand the influence to the pantograph head on the quality of contact.



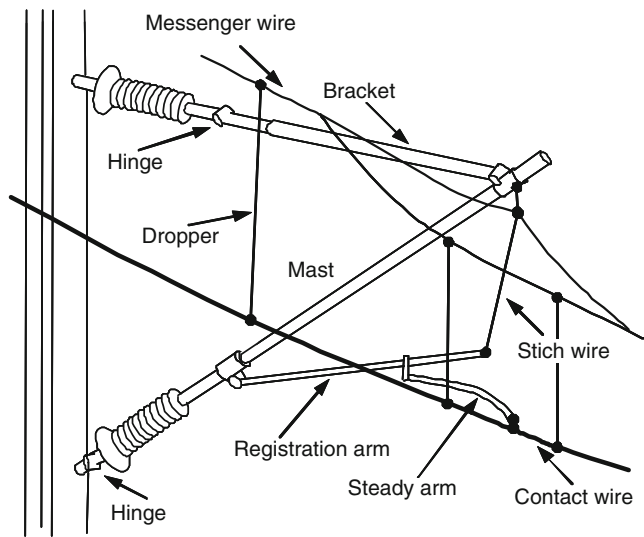


Fig. 11 Representation of a SNCF 25 kV suspended catenary

## 4.2 Catenary Finite Element Model

Catenaries are complex periodical structures, such as those presented in Fig. 11. Examples of typical structural elements involved in the catenary model are the contact, stitch and messenger wires, droppers and registration arms. Depending on the catenary system there are other elements that may have to be considered. In any case, the contact wire is the responsible for the contact between catenary and pantograph and therefore the element that provides electrical power. The messenger wire prevents excessive sag caused by the contact wire weight. Both of these wires are connected by vertical, tensile force droppers.

Even in a single European country there are different types of catenaries in use with different particularities in their construction. The contact wire is typically characterized by a small cross-section, compared to its length, being primarily suspended at the masts. Depending on the topology of the track and on the exposure to transversal winds the masts are placed at a distance of 27–63 m from each other. To maintain a constant mechanical stiffness of the contact wire a set of elements are designed to suspend the contact wire at these locations, specific of each suspended catenary type. In the French 25 kV catenary represented in Fig. 11 the contact wire is suspended by a low inertial elemental called the steady arm which is linked to the registration arm. The latter is suspended with respect to the messenger wire by the stitch wire and is connected by a hinge to the mast. This solution aims at limiting the dynamic coupling between the contact wire and the supporting elements. To minimize the spatial curve described by the contact wire and to maximize the wave propagation velocity of the contact wire a static load is applied to its extremities.

If seen from the top, the contact wire is suspended forming a *zigzag* around the longitudinal direction, designated by stagger. This geometric characteristic of the suspended catenary enables a constant wear of the pantograph registration strip.

### 4.3 Simulation Scenario and Results

To be able to understand the influence of the flexibility of the structural elements of the pantograph on the contact dynamics of the pantograph–catenary interface a single pantograph scenario is analyzed. The scenario corresponds to a single pantograph system attached to a railway vehicle running at approximately 300 km/h on a straight track, as depicted in Fig. 12.

The flexible multibody model 2 allows the analysis of the deformation of the upper arm. As expected the deformation is described by the bending modes of vibration. The results depicted in Fig. 13 show that the dominant mode of vibration on the pantograph top arm behavior is the first bending mode.

The bending of the upper arm results in lowering the position of the contact points with the pantograph head, as depicted in Fig. 14. However, the differences observed on the contact kinematics are not reflected on the contact forces, which are similar for the rigid and flexible models as seen in Fig. 15.

The analysis of the influence of pantograph head deformation in the contact force generated due to the pantograph–catenary interaction is analyzed also. Disregarding the deformation of the main frame can be, it is possible to understand the influence of the flexibility of the pantograph head by modeling the pantograph using the flexible multibody model 4. As depicted in Fig. 16, the first and second bending modes contribute to the deformation of the pantograph head. The deformation of the pantograph head has a very small influence on the contact forces, as seen in Fig. 17.

Although, for operational conditions and considering the present catenary model the influence of the deformation of the pantograph head may be disregarded without loss of accuracy, its influence is important to develop an actively controlled pantograph. Another aspect not studied in this work is the effect of the pantograph

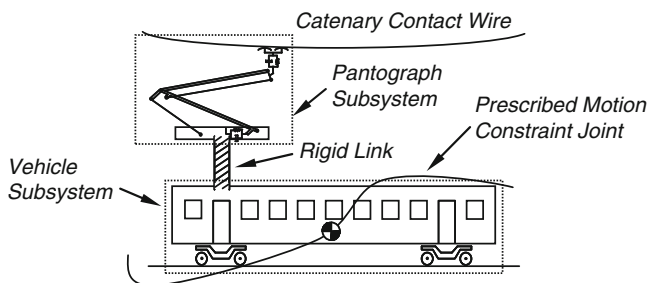


Fig. 12 Scenario for a high-speed train equipped with a pantograph running on a tangent track

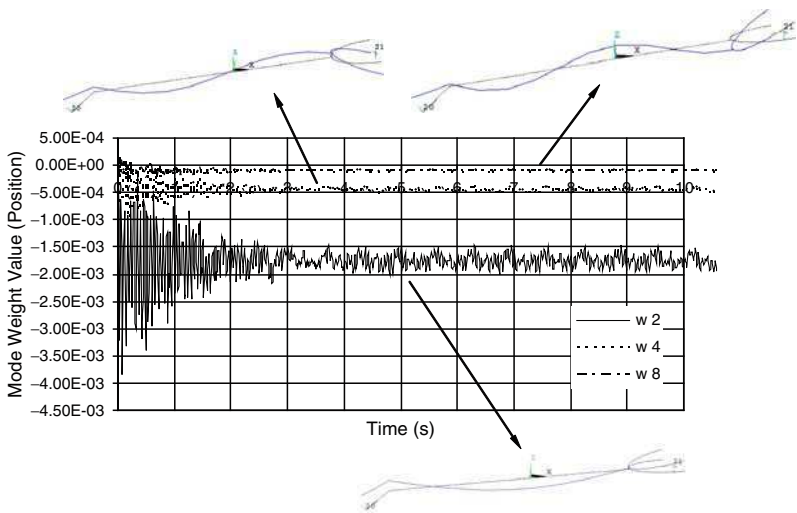


Fig. 13 Modal contribution of upper arm modes of vibration to the dynamic response of model 2

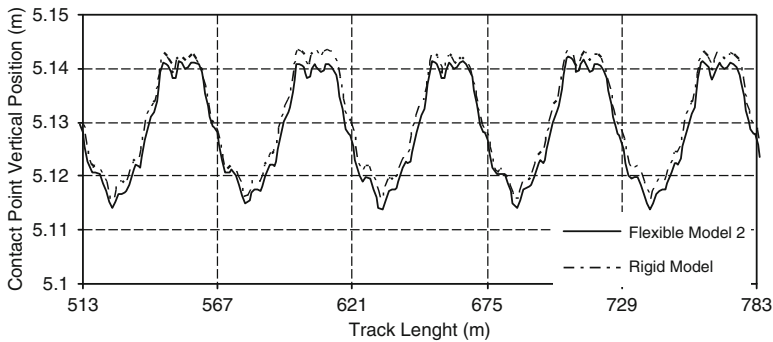


Fig. 14 Vertical position of the catenary contact point on the registration strip – Model 2

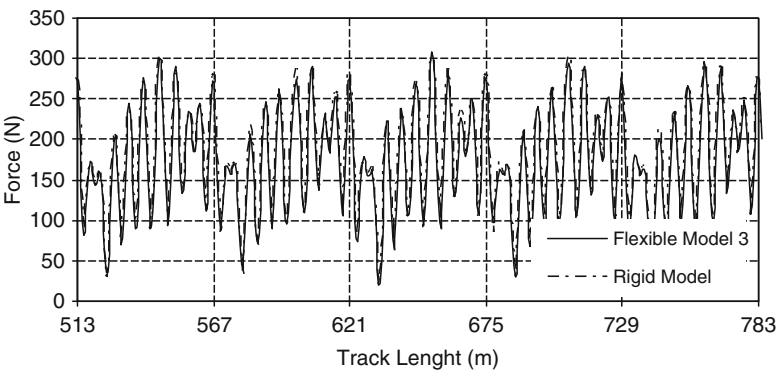
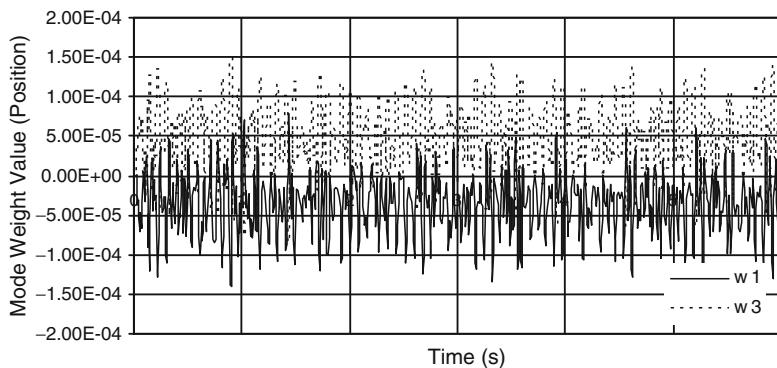
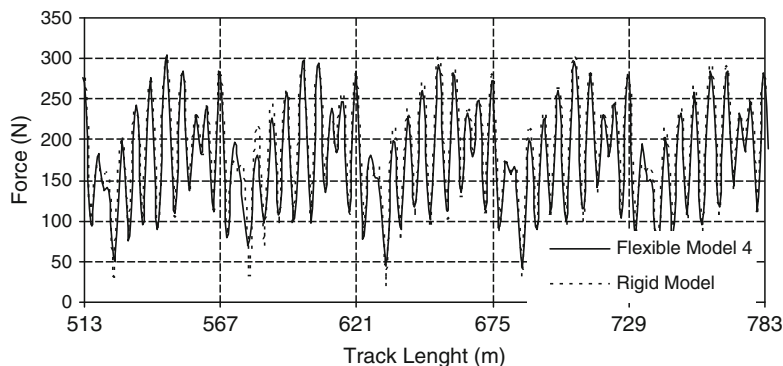


Fig. 15 Contact force (filtered at 20 Hz) for the rigid and flexible pantograph model 2





**Fig. 16** Modal contributions of the pantograph head deformation for model 4



**Fig. 17** Contact force results (filtered at 20 Hz) for the flexible pantograph model 4 versus rigid pantograph model, for the scenario of a train equipped with a single pantograph

flexibility in extreme conditions or when excited close to its natural frequencies, due to operational or defect conditions. It is expected that, under these conditions, the flexibility of the pantograph components cannot be disregarded.

## 5 Conclusions

The development of flexible multibody models of the pantograph is achieved in a straightforward way using the rigid multibody model as a base. The implementation of the virtual bodies methodology allows the definition of standard rigid kinematic joints, force elements or external applied non-linear forces to link flexible bodies. A pantograph model is devised testing the influence of the deformation of the most relevant bodies to the global dynamic behavior of the pantograph due to the

pantograph–catenary interaction. The deformation of main frame of the pantograph that reacts to low frequency solicitations of the contact force does not influence the numerical results. The kinematic relations of the main-frame lead to a canceling effect of the deformation of the lower arm and of the top arm. Furthermore, as expected, the pantograph head does not show a high level deformation, thus not influencing the contact force results, at least when comparing with the effects of other perturbations. It can be stated that for non-perturbed scenarios, for a pantograph running on a straight track at 300 km/h it is possible to disregard the effects due to the flexibility of the bodies. Nevertheless the use of flexible multibody models of the pantograph is important to be able to simulate the dynamic response of the pantograph–catenary system to defects, for example.

**Acknowledgement** The work presented has been developed in the framework of the European Project EUROPAC (European Optimized Pantograph Catenary Interface, contract no. STP4-CT-2005-012440) with the partners SNCF, Alstom Transport, ARTTIC, Banverket, České dráhy akciová společnost, Deutsche Bahn, Faiveley Transport, Mer Mec SpA, Politecnico di Milano, Réseau ferré de France, Rete ferroviaria italiana, Trenitalia SpA, UNIFE, Kungliga Tekniska Högskolan. The collaboration of SNCF, Faiveley Transport and Politécnico di Milano to the work reported is specially acknowledged. The support of Fundação para a Ciência e Tecnologia (FCT) through the grant SFRH/BD/18848/2004 is also gratefully acknowledged.

## References

1. Bocciolone M, Resta F, Rocchi D, Tosi A, Collina A (2006) Pantograph aerodynamic effects on the pantograph–catenary interaction. *Vehicle Sys Dyn* 44(S1):560–570
2. Pombo J, Ambrósio J, Pereira M, Rauter F, Collina A, Facchinetti A (2009) Influence of the aerodynamic forces on the pantograph–catenary system for high speed trains. *Vehicle Syst Dyn* 47(11):1327–1347
3. EUROPAC Project no. 012440 (2007) Modelling of degraded conditions affecting pantograph–catenary interaction. Technical Report EUROPAC-D22-POLI-040-R1.0, Politecnico di Milano, Milan, Italy
4. Song J, Haug EJ (1980) Dynamic analysis of planar flexible mechanisms. *Comput Methods Appl Mech Eng* 24:359–381
5. Shabana A (1982) Dynamic analysis of large-scale inertia variant flexible systems. PhD thesis, University of Iowa, Iowa City
6. Shabana A, Wehage R (1989) A coordinate reduction technique for transient analysis of spatial structures with large angular rotations. *J. Struct Mech* 11:401–431
7. Yoo WS, Haug EJ (1986) Dynamics of flexible mechanical systems using vibration and static correction modes. *ASME J Mech Trans Auto Design* 108:315–322
8. Wu S, Haug EJ (1988) Geometric non-linear substructuring for dynamics of flexible mechanical systems. *Int J Numer Methods Eng* 26:2211–2226
9. Chang B, Shabana A (1990) Nonlinear finite element formulation for large displacement analysis of plates. *ASME J Appl Mech* 57:707–718
10. Melzer F (1994) Symbolisch-numerische Modellierung Elastischer Mehrkörpersysteme mit Anwendung auf Rechnerische Lebensdauervorhersagen. PhD thesis, University of Stuttgart, Germany
11. Ambrósio J, Gonçalves J (2001) Complex flexible multibody systems with application to vehicle dynamics. *Multibody Syst Dyn* 6(2):163–182

12. Geradin M (1984) Finite element approach to kinematic and dynamic analysis of mechanisms using Euler parameters. In: Taylor C (ed.) *Numerical Methods for Non-linear Problems*, vol 2. Pineridge, Swansea
13. Geradin M, Cardona A, Doan DB, Duysens J (1995) Finite element modeling concepts in multibody dynamics. In: Pereira MS, Ambrósio J (eds) *Computer aided analysis of rigid and flexible multibody systems*. Kluwer, Dordrecht, The Netherlands, pp 233–284
14. Simo JC, Vu-Quoc L (1986) On the dynamics of flexible beams under large overall motions – the planar case: Part I. *ASME J Appl Mech* 53:849–854
15. Kane TR, Ryan RR, Banerjee AK (1987) Comprehensive theory for the dynamics of a general beam attached to a moving rigid base. *J Guidance Control Dyn* 10:139–151
16. Bathe K-J, Bolourchi S (1979) Large displacement analysis of three-dimensional beam structures. *Int J Numer Methods Eng* 14:961–986
17. Belytschko T, Hsieh BJ (1973) Nonlinear transient finite element analysis with convected co-ordinates. *Int J Numer Methods Eng* 7:255–271
18. Simo JC, Vu-Quoc L (1988) On the dynamics in space of rods undergoing large motions – a geometrically exact approach. *Comp Methods Appl Mech Eng* 66:125–161
19. Cardona A, Geradin M (1988) A beam finite element non-linear theory with finite rotations. *Int J Numer Methods Eng* 26:2403–2438
20. Cardona A, Geradin M (1991) Modelling of superelements in mechanism analysis. *Int J Numer Methods Eng* 32:1565–1594
21. Shabana A (1997) Definition of the slopes and the finite element absolute nodal coordinate formulation. *Multibody Syst Dyn* 1:339–348
22. Ambrósio J, Nikravesh PE (1992) Elastic-plastic deformations in multibody dynamics. *Non-linear Dyn* 3:85–104
23. Ambrósio J, Pereira M (1994) Flexibility in multibody dynamics with applications to crash-worthiness. In: Pereira MS, Ambrósio J (eds) *Computer aided analysis of rigid and flexible multibody systems*. Kluwer, Dordrecht, The Netherlands, pp 199–232
24. Ambrósio J, Ravn P (1997) Elastodynamics of multibody systems using generalized inertial coordinates and structural damping. *Mech Struct Mach* 25:201–219
25. Cavin RK, Dusto AR (1977) Hamilton's principle: finite element method and flexible body dynamics. *AIAA J* 15(12):1684–1690
26. Pereira M, Proença P (1991) Dynamic analysis of spatial flexible multibody systems using joint co-ordinates. *Int J Numer Methods Eng* 32:1799–1812
27. Nikravesh PE, Lin Y-S (2003) Body reference frames in deformable multibody systems. *Int J Multiscale Comput Eng* 1:1615–1683
28. Ambrósio J (2007) Flexible multibody systems with linear and nonlinear deformations. In: Flores P, Silva M (eds) *Proceedings of DSM2007 – Conferência Nacional de Dinâmica de Sistemas Multicorpo*, Guimarães, Portugal, 6–7 December 2007
29. Ambrósio J (2003) Efficient kinematic joint descriptions for flexible multibody systems experiencing linear and non-linear deformations. *Int J Numer Methods Eng* 56:1771–1793
30. Bae DS, Han JM, Choi JH (2000) A implementation method for constrained flexible multibody dynamics using virtual bodies and joint. *Multibody Syst Dyn* 4:207–226
31. Gonçalves J, Ambrósio J (2002) Advanced modeling of flexible multibody dynamics using virtual bodies. *Comput Assist Mech Eng Sci* 9(3):373–390
32. Gonçalves J (2002) Rigid and flexible multibody systems optimization for vehicle dynamics. PhD dissertation, Instituto Superior Técnico, Lisbon, Portugal
33. Gardou M (1984) Etude du comportement dynamique de l'ensemble pantographe-caténaire (Study of the dynamic behavior of the pantograph-catenary) (in French). PhD thesis, Paris, France
34. Jensen CN (1997) Nonlinear systems with discrete and discontinuous elements. PhD thesis, Technical University of Denmark, Lyngby, Denmark
35. Dahlberg T (2006) Moving force on an axially loaded beam – with applications to a railway overhead contact wire. *Vehicle Syst Dyn* 44(8):631–644

36. Labergri F (2000) Modélisation du Comportement Dynamique du Système Pantographe-Caténaire (Model for the Dynamic Behavior of the System Pantograph-Catenary) (in French). PhD thesis, Ecole Doctorale de Mécanique de Lyon, Lyon, France
37. Seo J-H, Sugiyama H, Shabana A (2004) Large deformation analysis of the pantograph-catenary systems. Technical Report #MBS04-7-UIC, Department of Mechanical Engineering, University of Illinois at Chicago, Chicago, Illinois
38. Seo J-H, Sugiyama H, Shabana A (2005) Modeling pantograph-catenary interactions for multibody railroad vehicle systems. In: Goicolea J, Cuadrado J, García Orden J (eds) Proceedings of the Multibody Dynamics 2005, ECCOMAS Thematic Conference, Madrid, Spain
39. Arnold M, Simeon B (2000) Pantograph and catenary dynamics: a benchmark problem and its numerical solution. *Appl Numer Math* 34(4):345–362
40. Veitl A, Arnold M (1999) Coupled simulations of multibody systems and elastic structures. In: Ambrósio J, Schiehlen W (eds) Proceedings of EUROMECH Colloquium 404 Advances in Computational Multibody Dynamics, Lisbon, Portugal, 20–23 September 1999, pp 635–644
41. Hulbert G, Ma Z-D, Wang J (2005) Gluing for dynamic simulation of distributed mechanical systems. In: Ambrósio J (ed) Advances on computational multibody systems. Springer, Dordrecht, The Netherlands, pp 69–94
42. Kubler R, Schiehlen W (2000) Modular simulation in multibody system dynamics. *Multibody Syst Dyn* 4:107–127
43. Lankarani HM, Nikravesh PE (1990) A contact force model with hysteresis damping for impact analysis of multibody systems. *AMSE J Mech Design* 112:369–376
44. Hughes T (1987) The finite element method: linear static and dynamic finite element analysis. Prentice-Hall, Englewood-Cliffs
45. Newmark NM (1959) A method of computation for structural dynamics. *J Eng Mech* 85:67–94
46. Augusta Neto M, Ambrósio J (2003) Stabilization methods for the integration of differential-algebraic equations in the presence of redundant constraints. *Multibody Syst Dyn* 10:81–105
47. Gear CW, Petzold L (1984) ODE methods for the solutions of differential/algebraic equations. *SIAM J Numer Anal* 21(4):716–728
48. Nikravesh P (1988) Computer-aided analysis of mechanical systems. Prentice-Hall, Englewood Cliffs
49. Rauter F, Pombo J, Ambrósio J, Chalansonnet J, Bobillot A, Pereira P (2007) Contact model for the pantograph-catenary interaction. *JSME Int J Syst Design Dyn* 1(3):447–457
50. Ambrósio J, Pombo J, Rauter F, Pereira M (2008) A memory based communication in the co-simulation of multibody and finite element codes for pantograph-catenary interaction simulation. In: Bottasso CL (ed) Multibody dynamics. Springer, Dordrecht, The Netherlands, pp 231–252
51. SNCF (2005) Numerical and experimental analysis of a pantograph (Analyse numérique et expérimentale d'un pantograph) (in French). Paris, France

# Maneuvering Multibody Dynamics: New Developments for Models with Fast Solution Scales and Pilot-in-the-Loop Effects

Carlo L. Bottasso, Giorgio Maisano, and Francesco Scorcelletti

**Abstract** The present paper focuses on trajectory optimization problems for multibody vehicle models, accounting for the presence of pilot-in-the-loop effects and fast dynamic components in the solution. The trajectory optimal control problem is solved through a direct approach by means of a novel hybrid single–multiple shooting method. Specific focus of the present work is the inclusion of pilot models in the optimization process, in order to improve the fidelity of the solution by considering the entire coupled human-vehicle system. In particular we investigate a series of maneuvers flown with helicopters, quantifying the performance loss due to human limitations of the pilot-vehicle system with respect to the sole vehicle case.

## 1 Introduction

The ability to simulate maneuvers of rotorcraft vehicles flying at the boundaries of their operating envelope is a valuable asset for performance analysis, handling qualities research, design and certification, pilot training, and support to the flight test activity. In general the maneuver of interest can be fully described in terms of quantities which should be minimized or maximized, subject to a variety of equality and inequality constraints [8–13]. Hence, one can usually give a precise mathematical definition of a maneuver by formulating an equivalent optimal control problem. The formulation of such a problem necessitates of a model of the vehicle system with its inputs, states and outputs, of a cost function and of a list of all constraints.

---

C.L. Bottasso (✉) and G. Maisano

Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano, Via La Masa 34,  
20156 Milano, Italy

e-mail: [carlo.bottasso@polimi.it](mailto:carlo.bottasso@polimi.it); [maisano@aero.polimi.it](mailto:maisano@aero.polimi.it)

F. Scorcelletti

Flight Mechanics Department, AgustaWestland, Via Giovanni Agusta 520,  
21017 Cascina Costa di Samarate (VA), Italy

e-mail: [francesco.scorcelletti@agustawestland.com](mailto:francesco.scorcelletti@agustawestland.com)

Clearly, the fidelity of the predictions made using this approach crucially hinges on the fidelity of the vehicle model. On the one hand, fidelity improvements may be obtained by considering a more sophisticated description of the vehicle; the current state-of-the-art calls for first-principle multibody models of the vehicle, coupling structural, fluid and servo fields. On the other hand, one might clearly consider the inclusion of a model of the pilot. In fact, in the absence of a pilot model, the solution of a trajectory optimization problem amounts to finding the limit performance trajectory flyable by a “perfect” pilot. In reality, the pilot is a complex system which can be modeled so as to account for sensory perceptions, learned behavior and biomechanical properties. Therefore, it is reasonable to assume that a maneuver optimized considering just a flight mechanics model of the vehicle will in general tend to overestimate the vehicle performance, as this has been computed without accounting for the limitations of various nature of a real pilot. To verify whether this is indeed the case, the present work tries to quantify this hypothesized performance loss due to the inclusion of a pilot model in the trajectory optimization process.

A human pilot model should account for various effects:

- **Sensorial perception:** the sensorial system of the pilot provides for a perception of movements, body position, accelerations, vibrations, etc., which enable the pilot to build a representation of the current situation.
- **Control behavior:** the pilot, based on the input provided by the sensorial information, evaluates the situation and, on the basis of a desired goal, elaborates a control law based on experience and training.
- **Command actuation:** the neuro-musculoskeletal system of the pilot acts like an actuator that takes as input the control law and translates it into movements of the vehicle controls (collective, cyclics, pedal).

In the literature, there is a wide range of pilot models which have been formulated for different applications. As suggested in [21], pilot models can be subdivided in the following categories:

- **Crossover Model:** a basic model for single-axis tracking tasks, which is useful for tuning more complete models. In the region of the open-loop crossover frequency, the product of the pilot transfer function and that of the vehicle is approximated as an integrator with time delay [26].
- **Isomorphic Models:** all models which try to explicitly approximate the dynamics of the human sensory and control systems. The *Structural Model* offers a simplified structural representation of the pilot dynamics in compensatory systems [20, 29]. Particular emphasis is given to sensorial feedback, which typically includes proprioceptive and vestibular feedbacks, while the neuro-muscular components of the model is approximated with a second order filter. The *Biophysical Models* give more emphasis to the dynamics of the pilot neuromuscular system [27]. Finally, *Biodynamic Models* are based on multibody dynamics approaches [22], and are used for investigating the effects of an accelerating/vibrating environment on the pilot control capabilities.
- **Algorithmic Models:** models whose principal focus is the control behavior of the pilot, but which may include some isomorphism achieving a good degree of

completeness. A typical example of this category is the *Optimal Control Model*, which considers the human pilot as an optimal controller [17], and where the sensorial component is taken into account by using a Kalman filter.

- **Behavioral Models:** models which consider the human pilot as a black box with nonlinear behavior. There are two principal approaches in this category: *Fuzzy-Logic Models*, which are based on fuzzy-set theory describing cause-and-effect relationships [25, 29], and *Neural Network Models*, which rely upon the capabilities of neural networks of accurately describing nonlinear input-output relationships, mapping pilot cues into control tasks [23].

Clearly, the most appropriate choice of a pilot model is strictly related to the particular application considered. In the framework of trajectory optimization, we need to account for all three aspects listed above, namely sensorial perception, control behavior, and command actuation. Furthermore, it would be preferable to work with a model formulated in state space form, so as to ease its integration in the overall maneuver optimal control problem.

The sensorial perception can be modeled by formulating appropriate observers, for example using Kalman filtering [17]. As a first step towards the goals set forth in this study, we have neglected this aspect of the problem in the present paper, although we plan on considering it in the continuation of this activity. In fact, although the inclusion of an observer in the maneuver optimal control problem formulation does not pose conceptual difficulties, we have postponed the modeling of this component of the pilot system because of the difficulty in finding data for the tuning of the filters.

The second aspect of the modeling, i.e. the control behavior, is in part already included in the formulation of a maneuver optimal control problem. In fact, the pilot elaborates a control law based on desired goals and constraints, which are in fact the very cost function and constraints which enter into the definition of the optimal control problem. However, some aspects of the control behavior are more subtle and difficult to model, such as for example the skills and experience of the pilot. Such effects are hard to model in precise mathematical terms, but we speculate here that they might be rendered through appropriate modifications of the cost function. For example, the modeling of piloting skills might account for degraded piloting behavior for maneuvers which require increased coordination and activity among the controls (increased workload) [15]. Such effects are easily included in the proposed maneuver optimal control approach, since the coding requires trivial modifications to the cost function routines. Nonetheless, specific experimental data are lacking, so that even in this case we have not considered these aspects in the present work, while waiting to perform experiments with pilots in a simulator to gather the observations necessary for the tuning of such models. Therefore, in this paper the control behavior is translated in the choice of a cost function that includes a problem-dependent goal quantity (e.g. altitude loss, time, etc.), and a control term which penalizes excessive control activity and/or excessive control rates; specific details on the choice of the cost functions are given below in the section of the paper devoted to the applications. Such modeling, although rather simple, probably captures a significant, and possibly the most significant, part of the pilot behavior.

The third aspect of the problem, the command actuation, can be modeled in a variety of ways. The more sophisticated approach is based on first-principle modeling of the musculoskeletal system using multibody dynamics, and typically includes rigid bodies with their inertial parameters, joints, muscles with their mechanical and physiological properties, interactional forces with the environment, and other components as required for the accurate representation of the real bio-system. Such level of detail is probably not necessary for capturing the effects of the limitations of the bio-system on the vehicle flight mechanics performance. Hence, a simpler approach is used here, where the effects of the musculoskeletal system are rendered in a global equivalent sense through the use of simple delay and filter models, as detailed below.

There are two principal approaches to the solution of trajectory optimization problem: indirect [16, 28] and direct methods [5–7, 10, 13]. Following our previous work [11], we prefer the direct approach even for the applications which are the focus of the present paper. In fact, in the case of the indirect methods one has first to derive the optimal control governing equations by using the calculus of variation, and then numerically solve the arising two-point boundary value problem. The manipulation of the vehicle equations of motion for deriving the optimal control governing equations makes it very hard or inefficient, if not altogether impossible, to use black-box flight simulators, where more often than not one does not have access to the source code. In the case of coupled vehicle-pilot models, the equations tend to become even more involved, so that here again the use of a direct approach allows for a simpler implementation. In fact, the direct approach does not require any manipulation of the equations, as one first discretizes the problem by time stepping (using either a transcription or a shooting method [11]) and then solves the resulting Non-Linear Programming (NLP) problem by a standard solver, such as SQP (Sequential Quadratic Programming).

Multibody vehicle models of rotorcraft systems include both slow flight mechanics scales and faster aero-elastic ones [14]. To treat more effectively this class of optimal control problems of multibody systems, we use multiple shooting on the flight mechanics scales, and single shooting on the faster aero-elastic ones; this avoids the enforcement of the multiple shooting gluing constraints for the faster scales, which greatly enhances convergence and in turn reduces the computational cost.

The paper is organized according to the following plan. At first, we describe the pilot model considered in this work and we present the equations of the coupled pilot-vehicle system. Secondly, we formulate in general terms the trajectory optimization problem. A discussion about the possible numerical solution strategies to solve this problem are given next; namely, we first describe the direct transcription approach and then we present the direct hybrid single–multiple shooting method. Finally, we investigate a number of maneuvering rotorcraft problems, and we assess the pilot-in-the-loop effects on the computed limit performance of the vehicle.



## 2 Coupled Pilot-Vehicle Model

As argued in the introduction, enriching a vehicle model by adding a pilot model is a way to improve the performance predictions made using trajectory optimization. The main task of a pilot is to govern the vehicle by deciding a suitable control law in relation with the maneuver goals, based on the current perception of the situation as provided by his/her sensory system. The optimal control model proposed in [24] and revisited in [17] is a possible way of rendering these effects. In the present work we adopt a similar approach, reformulating it in the context of trajectory optimization. This way, the decision level control behavior of the pilot can be considered as embedded in the objective function of the maneuver optimal control problem.

The two remaining aspects of human pilot limitations are due to sensorial perceptions and command actuation. As a first step towards the more ambitious goal of a complete pilot modeling system, we consider here a simple actuator pilot model (Fig. 1), in order to assess its impact on the vehicle performance predictions, as well as on the computational cost and robustness of the numerical procedures.

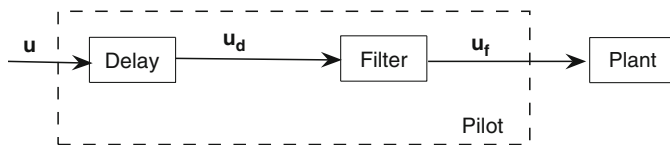
The vehicle equations of motion can be expressed as

$$\mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}, t) = 0, \quad (1)$$

where  $\mathbf{x}$  are the flight mechanics states, and  $\mathbf{u}$  the vehicle control inputs (collective, longitudinal and lateral cyclics, and pedal). More in general, rotorcraft multibody models are described in terms of differential–algebraic equations which also include Lagrange multipliers and constraint equations; however, in the sole interest of a lighter notation and simpler discussion, we consider in the following the ordinary differential set of equations expressed by (1).

The pilot actuator system is modeled using a pure time delay [17], operating in series with a second order filter for the neuromuscular element [29] for each control input. The pure time delay is approximated by a second-order Padè transfer function, which provides excellent accuracy over the frequency range of the pilot (0.1–10 rad/s) [17]; for the single channel we have:

$$Y_d(s) = \frac{1 - \frac{1}{2}(\tau s) + \frac{1}{8}(\tau s)^2}{1 + \frac{1}{2}(\tau s) + \frac{1}{8}(\tau s)^2}. \quad (2)$$



**Fig. 1** Pilot model: pure delay and second-order filter

The second-order filter [29] is written as

$$Y_f(s) = \frac{\omega_{NM}^2 s}{s^2 + 2\zeta_{NM}\omega_{NM}s + \omega_{NM}^2}. \quad (3)$$

The series of delay and filter on each control channel can be written in linear state space form as

$$\dot{\mathbf{x}}_p = \mathbf{A} \mathbf{x}_p + \mathbf{B} \mathbf{u}, \quad (4a)$$

$$\mathbf{u}_f = \mathbf{C} \mathbf{x}_p + \mathbf{D} \mathbf{u}, \quad (4b)$$

where  $\mathbf{x}_p$  is the neuromuscular state of length  $4n_u$ , where the number of control inputs  $n_u$  is equal to 4 for a rotorcraft vehicle. The elements of matrices  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$  and  $\mathbf{D}$  depend on the delay and filter parameters, and in particular on the time constant  $\tau$  of the pure delay, and on the damping factor  $\zeta_{NM}$  and undamped natural frequency  $\omega_{NM}$  of the open-loop neuromuscular system. Referring to Fig. 1, it should be noted that the inputs of the pure time delay module are the “desired” command pilot inputs  $\mathbf{u}$ , while the inputs of the neuromuscular module are the delayed command inputs  $\mathbf{u}_d$ . Finally, the delayed and filtered inputs  $\mathbf{u}_f$  actuate the rotorcraft vehicle model.

Collecting together (1) and (4), we can write the governing equations of the coupled pilot-vehicle model as

$$\dot{\mathbf{x}}_p = \mathbf{A} \mathbf{x}_p + \mathbf{B} \mathbf{u}, \quad (5a)$$

$$\mathbf{u}_f = \mathbf{C} \mathbf{x}_p + \mathbf{D} \mathbf{u}, \quad (5b)$$

$$\mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}_f, t) = 0. \quad (5c)$$

Formally, by collecting all states in a unique state vector  $\mathbf{x}_{pv}^T = (\mathbf{x}^T, \mathbf{x}_p^T)^T$ , by collecting all dynamic equations (5) into a single function  $\mathbf{f}_{pv}$  and eliminating all algebraic equations, we can write the governing equations of the coupled pilot-vehicle system in the following compact form:

$$\mathbf{f}_{pv}(\dot{\mathbf{x}}_{pv}, \mathbf{x}_{pv}, \mathbf{u}, t) = 0. \quad (6)$$

When optimizing a maneuver considering only the stand-alone vehicle model, one uses (1); on the other hand, when the pilot is included in the optimization the augmented system (6) is used. Formally, the two are identical, so that no changes are necessary to the trajectory optimization software for dealing with the coupled pilot-vehicle model.

### 3 Formulation of Maneuvers as Optimal Control Problems

A maneuver can be defined as a dynamic transition between two steady state (trimmed) configurations [18], although in the present context it is useful to give a looser interpretation of the term by considering also the case of terminal conditions

which are not trimmed. Clearly, given a starting and arrival configuration, there is an infinite number of ways to transition between the two. A possible way to remove this arbitrariness is to formulate a maneuver as a constrained optimal control problem [8–10, 13].

The maneuver optimal control problem requires the minimization or maximization of a cost or merit function (e.g. time, altitude loss, control activity, fuel consumption, etc.), which in general can be expressed in terms of the vehicle states or outputs and of the control inputs. Furthermore, the optimization problem is constrained by a number of conditions that should be met by the solution:

- First, the so-called compatibility conditions must be fulfilled at each time instant of the maneuver; in other words, it is required that the computed solution satisfies the equations of motion of a suitable flight mechanics model of the vehicle.
- Second, the solution should remain within the flight envelope and operational limits of the vehicle.
- Finally, most maneuvers of practical interest (Category-A, ADS-33, flare at the exit of an autorotation, etc.) are typically characterized by other equality and inequality constraints which need to be met in order to satisfy given performance and procedural requirements and that, collectively, contribute to giving a precise definition of the maneuver of interest.

The maneuver optimal control problem can be formally expressed as:

$$\min_{\mathbf{x}, \mathbf{y}, \mathbf{u}, T} \quad J = \phi(\mathbf{y}, t)|_0^T + \int_0^T L(\mathbf{y}, \mathbf{u}, \dot{\mathbf{u}}, t) dt, \quad (7a)$$

$$\text{s.t.:} \quad \mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}) = 0, \quad (7b)$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}), \quad (7c)$$

$$\mathbf{g}(\mathbf{y}, \mathbf{u}, t) \in [\mathbf{g}^{\min}, \mathbf{g}^{\max}]. \quad (7d)$$

Solving the problem consists in finding the control function  $\mathbf{u}(t)$ , and hence through (7b) and (7c) the associated functions  $\mathbf{x}(t)$  and  $\mathbf{y}(t)$ , which minimize the cost  $J$  given by (7a). In general, the cost includes a boundary quantity which accounts for values of the outputs at the initial and/or final instants, as well as an integral cost term. The problem is defined on the interval  $\Omega = [0, T]$ ,  $t \in \Omega$ , where the final time  $T$  is typically unknown and must be determined as part of the solution to the problem.

The model governing equations appear among the problem constraint conditions, and are expressed by (7b) and (7c), where  $\mathbf{x}$  are the states,  $\mathbf{u}$  the inputs and  $\mathbf{y}$  the outputs. As shown in the previous section, the model governing equations (7b) can be represented by (1) when considering the stand-alone vehicle model, or by (6) for the coupled pilot-vehicle case.

All maneuver-defining and/or envelope-protection constraints are expressed as generic algebraic non-linear constraints by (7d). These may include as special cases boundary (initial ( $t = 0$ ) and/or terminal ( $t = T$ )) conditions, constraints at

unknown internal time events ( $t = T_i$ ), generic constraints defined over the whole maneuver duration ( $t \in [0, T]$ ), which clearly may also include, as it is often the case in practical applications, simple bounds on the inputs and states/outputs.

## 4 Direct Solution of Maneuver Optimal Control Problems

As discussed in [11], the direct approach is often the preferable way to solve the optimal control problem (7), for a series of practical advantages with respect to the classical indirect method. According to the direct approach, the optimal control problem is first discretized and subsequently optimized. This procedure yields a discrete parameter optimization or NLP problem [19], which can be written as

$$\min_{\mathbf{z}} K(\mathbf{z}), \quad (8a)$$

$$\text{s.t. } \mathbf{a}(\mathbf{z}) = \mathbf{0}, \quad (8b)$$

$$\mathbf{b}(\mathbf{z}) \in [\mathbf{b}^{\min}, \mathbf{b}^{\max}], \quad (8c)$$

where  $\mathbf{z}$  is a vector of algebraic unknowns, and  $K$  is a scalar objective function which represents an approximation of the cost  $J$  of (7a). The equality constraints (8b) are generated by the discretization of the equations of motion (7b,7c), while the inequality constraints (8c) by all other maneuver-defining constraints (7d). Notice that the problem defined by (8) is characterized by unknown algebraic parameters  $\mathbf{z}$ , while the optimal control problem (7) by functional unknowns.

The specific form of the vector of algebraic unknowns and of the constraints in problem (8) depends on the method used for performing the discretization. Our software program TOP (Trajectory Optimization Program) [11] implements both the direct transcription and the direct multiple shooting methods, which are briefly reviewed next.

### 4.1 Direct Transcription

This method is very effective and robust, but it is typically applicable only to models which have low-moderate complexity [13], i.e. which do not have solution time scales which are too fast with respect to the overall maneuver duration, and/or do not possess too large a number of states.

The time interval  $\Omega$  is partitioned as  $0 = t_0 < t_1 < \dots < t_N = T$ , where the generic time element is  $\Omega^n = [t_n, t_{n+1}]$ ,  $n = (0, N - 1)$ , of time step size  $h^n = t_{n+1} - t_n$ . On each time element  $\Omega^n$ , the governing equations (7b) are discretized using a suitable numerical method. The resulting discrete equations are expressed as

$$\mathbf{f}_h(\mathbf{x}_{n+1}, \mathbf{x}_n, \mathbf{u}^n, h^n) = \mathbf{0}, \quad n = (0, N - 1), \quad (9)$$

where  $\mathbf{f}_h$  is an algorithmic approximation of function  $\mathbf{f}$  of (7b),  $\mathbf{x}_n, \mathbf{x}_{n+1}$  are the values of the state vector at  $t_n$  and  $t_{n+1}$ , respectively, while  $\mathbf{u}^n$  represents the value of the control vector within the step. In general there might be additional internal stages for both the state and the control variables, depending on the numerical method [11].

The NLP problem (8) is defined as follows. First, the NLP vector of parameters is chosen as:

$$\mathbf{z} = (\mathbf{x}_{n=(0,N)}, \mathbf{u}^{n=(0,N-1)}, T)^T, \quad (10)$$

i.e. it is defined by the discrete states and control values on the computational grid, and the final time. Notice that, if one needs a very large number of time steps to accurately resolve the solution, the size of  $\mathbf{z}$  will be large, up to the point of making this approach unsuitable in terms of computational burden.

Next, the cost  $J$  of (7a) is discretized in terms of  $\mathbf{z}$  as given by (10), obtaining the discrete cost  $K$  of (8a). Then, the discretized ODEs within each step, (9), become the set of NLP equality constraints appearing in (8b). Finally, all other problem constraints and bounds, (7d), are expressed in terms of the NLP variables  $\mathbf{z}$  and become the NLP inequality constraints of (8b).

## 4.2 Direct Multiple and Hybrid Single–Multiple Shooting

Multiple shooting is typically used in applications of moderate/high complexity, i.e. with solution time scales which are fast with respect to the maneuver duration, and/or a moderate/large number of degrees of freedom [13].

The time domain  $\Omega$  is partitioned as  $0 = t_0 < t_1 < \dots < t_M = T$  with  $\Omega^m = [t_m, t_{m+1}]$ ,  $m = (0, M-1)$ , where each  $\Omega^m$  is a shooting segment. In each shooting segment  $\Omega^m$ , the controls are discretized as  $\mathbf{u}^m(t) = \sum_{i=1}^{N_c^m} s_i(t) \mathbf{u}_i^m$ , where  $s_i(t)$  are basis functions, in particular cubic splines in the present implementation, and  $\mathbf{u}_i^m$  are  $N_c^m$  unknown discrete control values. The control approximations are confined on each shooting segment; this has the effect of decreasing the computational cost of finite differencing by increasing the problem sparsity. Constraints are enforced at the shooting segment boundaries to guarantee the continuity of the controls up to  $C^1$ .

In the case of direct multiple shooting, the NLP problem (8) is defined as follows. First, the NLP unknown parameters are chosen as:

$$\mathbf{z} = (\mathbf{x}_{m=(0,M)}, \mathbf{u}_{i=(1,N_c^m)}^{m=(0,M-1)}, T)^T, \quad (11)$$

i.e. they represent the discrete values of the states at the interfaces between shooting segments, the discrete values of the controls within each segment, and the final time.

Next, the governing ODEs (7b) are marched in time within each shooting segment  $\Omega^m$ , starting from the initial conditions provided by the values of the states  $\mathbf{x}_m$  at the left boundary of the segment. The effect of the forward integration is to generate a discrete time history of states within  $\Omega^m$ , which we label  $\mathbf{x}_i^m$ ,  $i = (1, N^m)$ , where  $N^m$  is the number of steps taken in that segment. The last value of this

sequence is named  $\tilde{\mathbf{x}}_{m+1} = \mathbf{x}_{N^m}^m$ , and represents the new estimate of the state variables at the right boundary of the shooting segment. Segments are then glued together by imposing the following equality constraints

$$\mathbf{x}_m - \tilde{\mathbf{x}}_m = 0, \quad m = (2, M). \quad (12)$$

Multiple shooting segments are used for stabilizing the forward integration of the vehicle equations of motion [4]. This is particularly important when analyzing unstable systems, which is often the case when considering rotorcraft vehicles.

Notice that the size of the unknown parameter vector  $\mathbf{z}$  is unrelated to the time step size used for marching the equations of motion within shooting arcs; hence, one may use very fine temporal discretizations without impacting the overall problem size, which in fact enables the solution of problems with a higher degree of complexity than in the direct transcription case [13].

In the direct multiple shooting case, the cost  $J$  of (7a) is discretized in terms of  $\mathbf{z}$  as given by (11) and evaluated using the segment time histories  $\mathbf{x}_i^m$ ; this yields the discrete cost  $K$  of (8a). Next, the gluing conditions (12) are used to express the set of NLP equality constraints appearing in (8b). All other problem constraints and bounds, (7d), are expressed in terms of the NLP variables  $\mathbf{z}$  and become the NLP inequality constraints of (8b).

For complex multibody systems denoted both by slow and fast solution components, we have observed that the satisfaction of the multiple shooting gluing constraints can be particularly difficult and usually ends up dominating the problem. Once again a rotorcraft multibody model provides for an excellent illustration of such difficulties. In fact, models have flight mechanics states which describe the gross rigid body motion of the vehicle through the position, orientation, linear and angular velocities of a body-attached (or floating, in the case of a flexible fuselage) frame of reference, as well as fast scales which are typically related to the rotor degrees of freedom, and include rigid and flexible blade states and aerodynamic states.

Often, a naive implementation of multiple shooting fails to achieve convergence for such complex multi-scale models. This is not surprising, since the rotor generates most of the aerodynamic forces acting on the vehicle and even small variations in its states may imply large variations in the resulting forces, which hinders the satisfaction of the gluing constraints.

We have found that these problems can be alleviated by using multi-time scale arguments [14]. In fact, the rotor states (both structural and aerodynamic) are significantly faster than the flight mechanics ones. Thus, since the multiple shooting treatment of these fast states is the main cause of the two aforementioned issues, i.e. raise in computational cost and difficulty in satisfying gluing constraints, one can think of treating slow and fast scales using different methods.

More specifically, a multiple shooting approach is used for the slow states. This is crucial, since with single shooting small changes early in the trajectory can produce dramatic effects at the end of it [4]; clearly, the problem is exacerbated when analyzing unstable systems. Hence, the multiple shooting treatment of slow scales avoids the blow up of the solution.

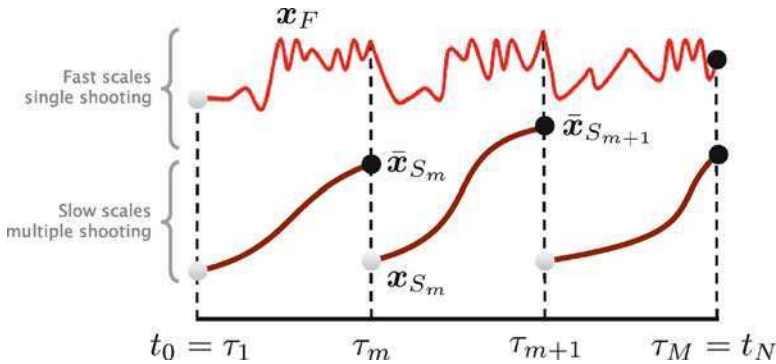


Fig. 2 Hybrid single-multiple shooting approach

On the contrary, fast scales are treated using a single shooting approach, as depicted in Fig. 2. This does not compromise the robustness of the procedure, since fast scales will not diverge if slow ones do not; hence, the stabilizing effect produced by the multiple shooting treatment of slow scales is felt also at the level of the fast ones.

With such a hybrid single-multiple shooting approach, the size of the resulting NLP problem is substantially reduced and so is the total computational cost. Furthermore, there are no gluing constraints to be enforced for the fast rotor states, since only the slow states need to be glued together at the shooting interfaces. This has the effect of greatly increasing the robustness of the procedure, and the convergence speed.

## 5 Applications and Results

In this section we consider the solution of maneuver optimal control problems of practical interest. We analyze the ADS-33 Lateral Reposition Mission Task Element (MTE) [3] for handling qualities assessment, as well as a Category-A fly-away [2]. Goal of these two examples is a first preliminary assessment of the effects of the inclusion of the simplified pilot model described earlier on in this work with respect to the computed limit performance.

The helicopter model, implemented using the rotorcraft multibody FLIGHTLAB code [1], represents a generic medium-size multi-engine four-bladed utility vehicle in the 9 ton class.

### 5.1 Lateral Reposition MTE

The ADS-33E-PRF specification [3] for military rotorcraft defines a series of MTEs which provide a basis for an overall assessment of the vehicle ability to perform

certain critical tasks, and result in an assigned level of handling qualities according to the Cooper–Harper rating scale. Each MTE is related to a maneuver that shall be accomplished considering specific constraints, as described in [3]. In fact, it is possible to formulate each MTE as a constrained optimal control problem [12]. Hence, with a software implementation of the procedures discussed in this work, it is possible to readily compile a library of MTEs of interest in order to predict the handling qualities characteristics of a specific rotorcraft.

We analyze here the Lateral Reposition MTE, considering both the stand-alone vehicle model and the coupled pilot-vehicle system.

According to the Lateral Reposition MTE [3], the helicopter, initially in hover, is supposed to translate laterally for 400 ft and then recover the initial hover configuration. The maneuver must be flown in ground effect since the initial and final positions are characterized by an altitude of 35 ft (the rotor diameter is 30 ft); altitude variations must be within  $\pm 10$  ft. Referring to Fig. 3, the maximum allowed displacement in the longitudinal direction is  $\pm 10$  ft, while the maximum heading misalignment is  $\pm 10$  deg with respect to the initial direction. The maneuver must be completed within 18 s.

One possible formulation of this MTE is to consider the following minimum time cost function (13):

$$J = T + \frac{1}{T} \int_0^T \dot{\mathbf{u}} \cdot \mathbf{W} \dot{\mathbf{u}} dt. \quad (13)$$

The first term enforces the minimum time condition, while  $\mathbf{W} = \text{diag}(\mathbf{w}_i)$  is a diagonal matrix of tunable weighting factors which penalize the control rates.

It is also necessary to constrain the vehicle trajectory so as to express the MTE path requirements described above. With this formulation of the problem the time constraint is not explicitly enforced, but verified a posteriori. In other words, one tunes the weight parameters  $\mathbf{W}$  in the merit function (13), this way controlling the aggressiveness of the maneuver. Then, once a solution has been computed, one verifies whether the maneuver was rapid enough and effectively completed within the maximum allotted time. Obviously there are limitations in the maneuver aggressiveness related to the vehicle capabilities and its flight envelope constraints. In this case the trajectory constraints are imposed directly through bounds on the position variables and heading angle:

$$|\psi(t)| \leq 10 \text{ deg}, \quad (14a)$$

$$|x(t)| \leq 10 \text{ ft}, \quad (14b)$$

$$|\Delta z(t)| \leq 10 \text{ ft}, \quad (14c)$$

$$0 \leq y(t) \leq 400 \text{ ft}. \quad (14d)$$

We solved this problem initially without considering a pilot model; once the “pilot-off” solution had been evaluated, we used it as the initial guess for the evaluation of the “pilot-on” case. The following values for the pilot actuator model were used [29]:  $\tau = 0.2$  s,  $\omega_{NM} = 10$  rad/s,  $\zeta_{NM} = \cos(\pi/4)$ .



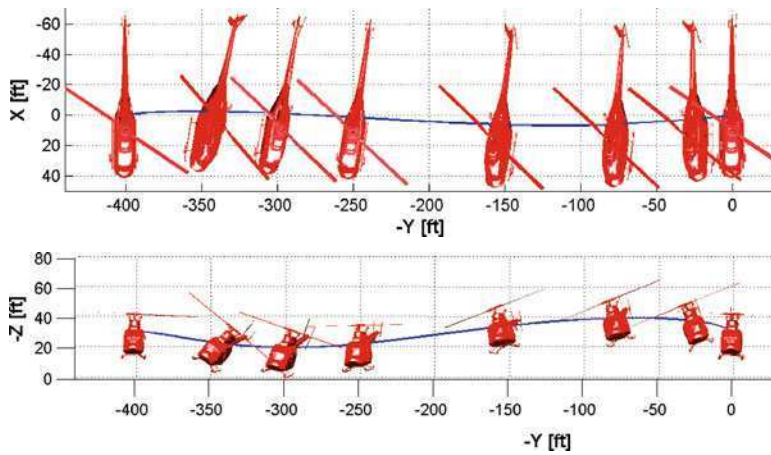


Fig. 3 Lateral Reposition MTE: snapshots

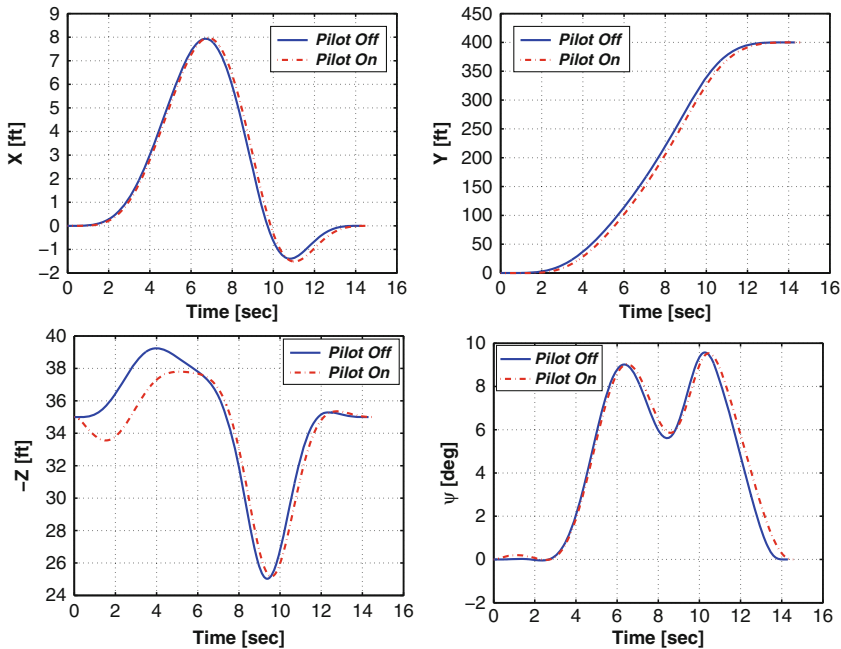
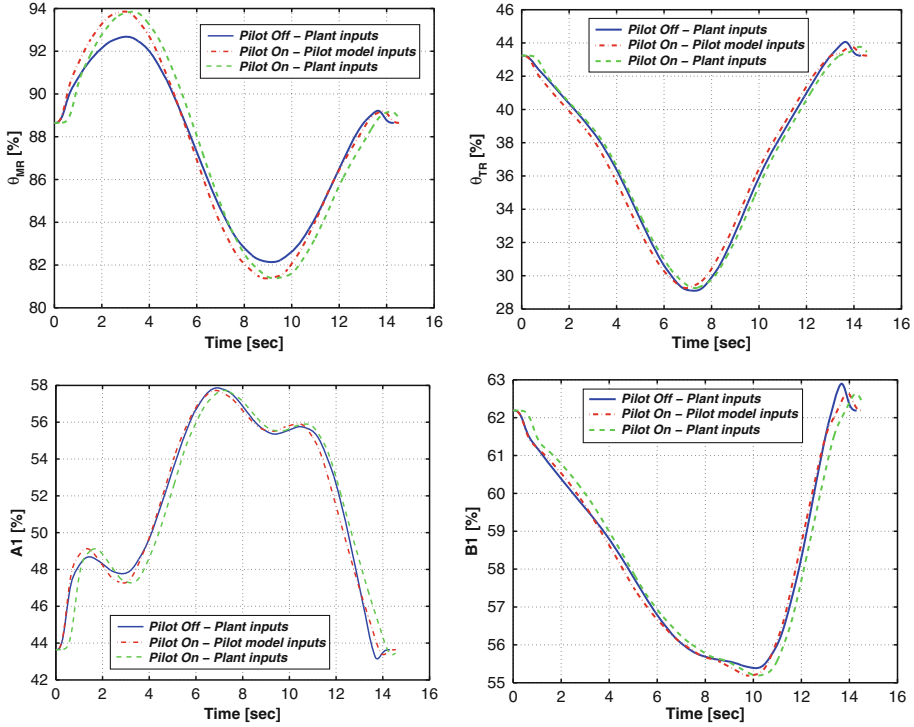


Fig. 4 Lateral Reposition MTE: X, Y, Z positions, and heading angle (from *top* to *bottom*, *left* to *right*)

Figure 3 shows some snapshots of the helicopter during the maneuver. Figure 4 gives the time histories of the constrained path variables, for both the pilot-off (solid lines) and the pilot-on cases (dash-dotted lines). Figure 5 shows the control time history; in the pilot-on case, both the computed pilot model inputs (vector  $\mathbf{u}$  in (5)) and the plant inputs (vector  $\mathbf{u}_f$  in (5)) are shown in the figure.



**Fig. 5** Lateral Reposition MTE: collective, pedal, lateral and longitudinal cyclic (from *top* to *bottom*, *left* to *right*)

The pilot-in-the-loop effects do not appear to generate significant differences with respect to the stand-alone vehicle model for both the trajectory and the control inputs. The maneuver duration is in both cases less than the 18 s prescribed by the normative, with a slightly longer total time for the pilot-on case. Both trajectory and controls do not appear to have been significantly affected by the neuromuscular lag.

## 5.2 Category-A Fly-Away

The effect of pilot actuation are investigated also for the case of a fly-away maneuver under Category A certification requirements [2]. A meaningful simulation policy for such a maneuver consists in the minimization of the altitude loss, according to the cost

$$J = H(T) + \frac{1}{T} \int_0^T \dot{\mathbf{u}} \cdot \mathbf{W} \dot{\mathbf{u}} dt, \quad (15)$$

The initial condition is a hover. A latency period of 1.2 s after the engine failure is taken into account, during which the pilot realizes the situation and there is no control activity. The power loss is modeled as

$$P_{av}(t) = P_H + (P_{OEI} - P_H) K^+(t) + P_H K^-(t), \quad (16a)$$

$$K^+(t) = \text{sca}(t - t_0)(1 - e^{-t/\tau^+}), \quad (16b)$$

$$K^-(t) = \text{sca}(t - t_0)e^{-(t-t_0)/\tau^-}, \quad (16c)$$

where  $t_0$  is the instant of engine failure,  $P_H$  is the hover power,  $P_{OEI} = 1750$  HP is the one engine inoperative maximum take-off power available, while  $\tau^+ = 2/9$  s and  $\tau^- = 1/9$  s are suitable time constants. An inequality constraint in the maneuver optimal control problem (7d) is used for ensuring that the power generated by the engine is at all times less than the available one, as expressed by (16a). The final conditions are

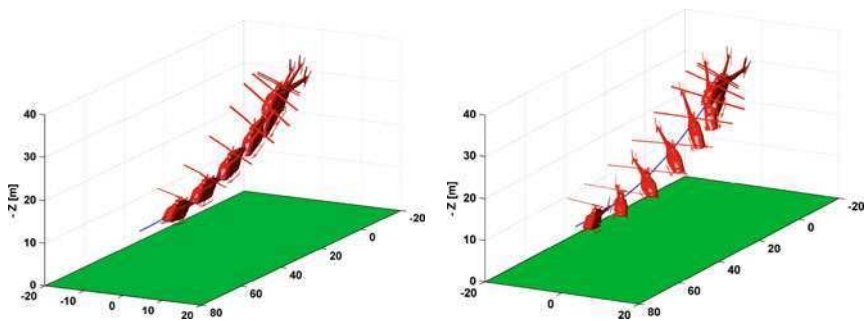
$$W(T_f) = 0 \text{ m/s}, \quad (17a)$$

$$p(T_f) = q(T_f) = r(T_f) = 0 \text{ deg/s}, \quad (17b)$$

$$\Omega(T_f) \geq 90\%. \quad (17c)$$

All simulations were conducted outside of ground effect. This single-phase formulation of the problem considers only the first part of the maneuver, i.e. from the engine loss to the moment the lowest point in the trajectory is reached. A multi-phase formulation of the same problem covering also the climb part of the Category-A maneuver was considered in [8–10].

The standard procedure is to fly this emergency maneuver in the longitudinal plane of the helicopter. In fact, simulations of this maneuver are often conducted with a two-dimensional helicopter model. However, using a three-dimensional model, one may observe that the solution converges to a three-dimensional maneuver with significant yaw and roll (see Fig. 6). The three-dimensional (3D) optimal maneuver altitude loss  $\Delta H_{\min}^{3D} = 15.77$  m improves on the two-dimensional (2D)



**Fig. 6** Category A, fly-away: optimal two-dimensional (*left*) and three-dimensional (*right*) trajectories, pilot-off case

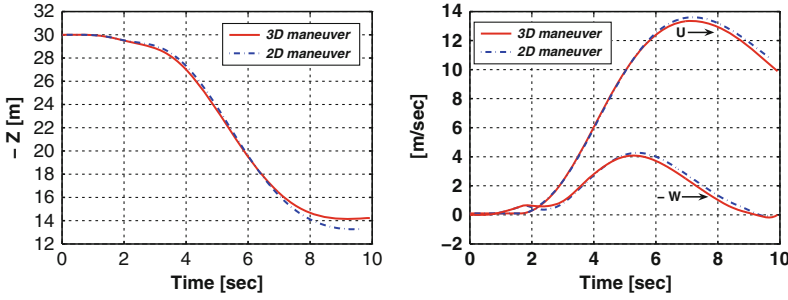


Fig. 7 Category A, fly-away: altitude loss comparison (*left*) and inertial velocities (*right*), pilot-off case

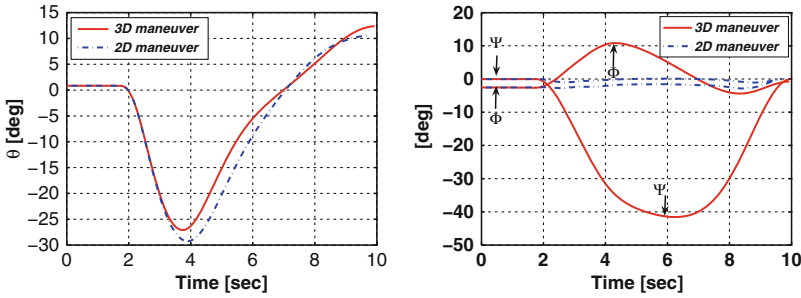


Fig. 8 Category A, fly-away: helicopter attitude, pilot-off case

optimal altitude loss  $\Delta H_{\min}^{2D} = 16.72$  m of about one meter (Fig. 7 at *left*), as confirmed by the opinion of test pilots. This gain can be explained by observing Fig. 8. In the 3D maneuver, both the roll and yaw angles increase and reach their respective maxima approximatively halfway throughout the maneuver. This attitude allows for some reduction in the vertical velocity (Fig. 7 at *right*), which explains the decreased altitude loss. Clearly, the control activity on the pedal and lateral cyclic is higher for the 3D maneuver than for the 2D one.

These two simulations were repeated including the pilot model. To simplify convergence, we used a bootstrapping procedure. The first guess was initialized to the solution computed without pilot model. Next, the control time histories of the guess solution were used for evaluating the pilot model dynamic constraints, thus obtaining initial estimates of the pilot state time histories. The stand-alone vehicle solution augmented with the pilot state time histories was then used as initial guess for the pilot-in-the-loop optimization.

For the coupled pilot-vehicle problem, the resulting optimal maneuvers do not change significantly in terms of control input profiles with respect to pilot-off simulations, but the altitude loss increases for both the 2D and 3D cases (see Fig. 9). In the 2D maneuver the altitude loss is  $\Delta \hat{H}_{\min}^{2D} = 18.56$  m with a difference of 1.84 m (10.38 %) with respect to the pilot-off case, while in the 3D case we obtain  $\Delta \hat{H}_{\min}^{3D} = 17.62$  m with a difference of 1.85 m (11.73 %). This is not simply due to

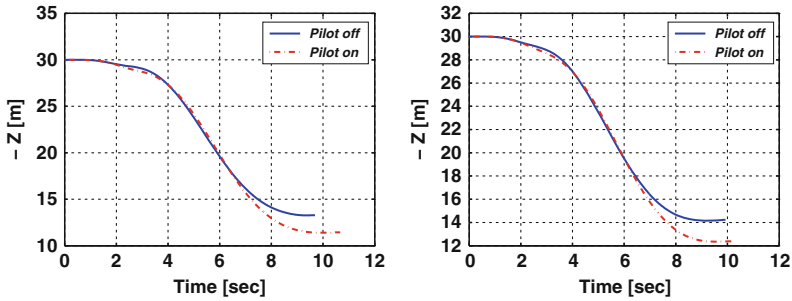


Fig. 9 Category A maneuver, fly-away: altitude loss comparison (*left*: 2D; *right*: 3D)

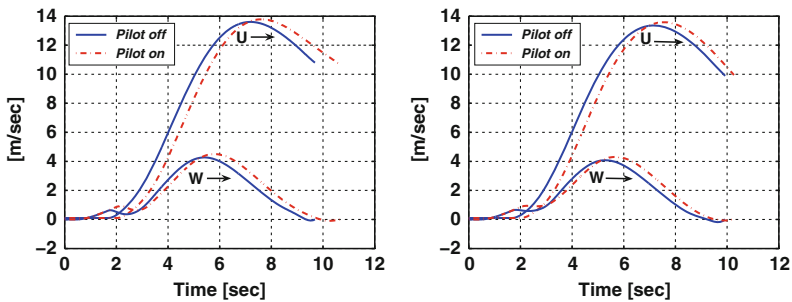


Fig. 10 Category A maneuver, fly-away: inertial velocities (*left*: 2D; *right*: 3D)

the fact that all time histories are delayed. The principal reason appears to be the delay in the pilot first reaction to the engine loss, which gives a higher maximum vertical velocity value, as shown in Fig. 10.

In conclusion, the introduction of a pilot model seems to have a non negligible effect on the performance estimation, which would seem to motivate further refinements in the simplified pilot model considered in this preliminary study. Furthermore, it appears that a 3D maneuver gives better performance (less altitude loss), than the usual 2D one. However, the 3D maneuver is harder to fly since it requires good coordination skills. Moreover, the pronounced sideslipping might make it difficult for the pilot to hold the visual references.

## 6 Concluding Remarks

In this work we have formulated a trajectory optimization approach to maneuver modeling in rotorcraft flight mechanics, including pilot-in-the-loop effects and fast dynamic solution components.

The formulation can accommodate the pilot control behavior as part of the definition of the cost function (and in this sense falls within the category of optimal control pilot models), as well as the command actuation and sensorial perception

aspects. In this work, the command actuation was rendered using global equivalent models through the use of a simple delay in series with a second order filter. Although a more sophisticated model, as for example a biomechanical multibody model, could be readily implemented in the formulation without conceptual difficulties, the present implementation is probably sufficient for capturing the relevant command actuation effects on the flight mechanics characteristics of the response. The sensorial perception component of the model was not considered here, mainly for the lack of sufficiently reliable data for the tuning of the required Kalman-based observers; this aspect of the problem is currently under investigation, and will be reported in a forthcoming publication. It is reasonable to speculate that the inclusion of the perception system model will determine further degradation of the performance, although the actual quantification of this aspect remains to be seen.

Based on the current state of this study, the following conclusions may be drawn:

- The performance degradation due to pilot-in-the-loop effects depends on the particular maneuver considered. In particular, it appears that for the Lateral Reposition MTE the pilot model induces negligible differences, while the Category-A rejected take-off shows a more pronounced effect with an increased altitude loss. Other maneuvers will be considered in the continuation of the present study.
- The inclusion of a pilot model in the optimal control formulation does not imply substantial difficulties, since the coupled pilot-vehicle system is formally identical to a generic vehicle model expressed in non-linear state space form.
- The current version of the pilot model has only a modest impact on the computational cost of the optimization, so that the code retains its ability to conduct complete maneuver simulations in the order of minutes on standard desktop computers.
- As for all optimization problems, better performance and robustness of the procedures relies also on good initial guesses of the solution, which in this case also requires initial estimates of the internal pilot states. This was achieved here using a bootstrapping procedure, based on an initial solution computed without pilot model, followed by the initialization of the pilot states obtained with the computed pilot-off control inputs. This procedure proved to be easy to implement and very effective.

**Acknowledgements** The present research is supported by AgustaWestland through a grant with the Politecnico di Milano, Marco Cicalè being the main project monitor. Simulations using the FLIGHTLAB code were conducted at the AgustaWestland headquarters in Cascina Costa, Italy, using AgustaWestland licenses. The contribution of C. Ravaioli and A. Ragazzi in the preparation of the examples is gratefully acknowledged.

## References

1. Advanced Rotorcraft Technology, Inc., 1685 Plymouth Street, Suite 250, Mountain View, CA 94043. <http://www.flightlab.com>
2. Anonymous (1999) Advisory circular 29-2C, certification of transport category rotorcraft. Federal Aviation Administration, Department of Transportation

3. Anonymous (2000) Handling qualities requirements for military rotorcraft. Aeronautical Design Standard, U.S. Army Aviation and Missile Command, Aviation Engineering Directorate, Rept. ADS-33E-PRF, Redstone Arsenal, AL
4. Ascher UM, Mattheij RMM, Russell RD (1995) Numerical solution of boundary value problems for ordinary differential equations. *Classics in Applied Mathematics*, SIAM, vol 13, Philadelphia
5. Betts JT (2001) Practical methods for optimal control using non-linear programming. SIAM, Philadelphia
6. Betts JT (1998) Survey of numerical methods for trajectory optimization. *J Guid Cont Dynam* 21(2):193–207
7. Bottasso CL, Croce A (2004) Optimal control of multibody systems using an energy preserving direct transcription method. *Multibody Syst Dyn* 12:17–45
8. Bottasso CL, Croce A, Leonello D, Riviello L (2005) Optimization of critical trajectories for rotorcraft vehicles. *J Am Helicopter Soc* 50:165–177
9. Bottasso CL, Croce A, Leonello D, Riviello L (2005) Rotorcraft trajectory optimization with realizability considerations. *J Aerospace Eng* 18:146–155
10. Bottasso CL, Chang C-S, Croce A, Leonello D, Riviello L (2006) Adaptive planning and tracking of trajectories for the simulation of maneuvers with multibody models. *Comput Meth Appl Mech Eng* 195:7052–7072
11. Bottasso CL, Maisano G, Scorcelletti F (2008) Trajectory optimization procedures for rotorcraft vehicles, their software implementation and applicability to models of varying complexity. American Helicopter Society 64th Annual Forum, Montréal, Canada. Also: *J Am Helicopter Soc*, under review
12. Bottasso CL, Scorcelletti F, Maisano G, Cicalè M, Ragazzi A (2008) Mission task elements and critical maneuvers simulation for rotorcraft vehicles. *Rotorcraft Handling Qualities Conference*, University of Liverpool, Liverpool, UK
13. Bottasso CL (ed.) (2008) Solution procedures for maneuvering multibody dynamics problems for vehicle models of varying complexity. *Multibody dynamics – computational methods and applications*, Computational Methods in Applied Sciences, ISBN 978-1-4020-8828-5, Springer, Dordrecht, The Netherlands
14. Bottasso CL, Maisano G (2009) Efficient rotorcraft trajectory optimization using comprehensive vehicle models by improved shooting methods. 35th European Rotorcraft Forum, Hamburg, Germany
15. Bradley R, MacDonald CA, Buggy TW (2005) Quantification and prediction of pilot workload in the helicopter/ship dynamic interface. *J Aerospace Eng* 219(5):29–443
16. Bryson AE, Ho YC (1975) *Applied optimal control*. Wiley, New York
17. Davidson JB, Schmidt DK (1992) Modified optimal control pilot model for computer-aided design and analysis. NASA, TM 4348, NASA Langley Research Center, Hampton, VA, USA
18. Frazzoli E (2001) Robust hybrid control for autonomous vehicle motion planning. Ph.D. Thesis, Department of Aeronautics and Astronautics. Massachusetts Institute of Technology, Cambridge, MA, USA
19. Gill PE, Murray W, Wright MH (1981) *Practical optimization*. Academic Press, London and New York
20. Hess RA (1997) Unified theory for aircraft handling qualities. *J Guid Control Dynam* 20(6):1141–1148
21. Hess RA (2006) Simplified technique for modelling piloted rotorcraft operations near ships. *J Guid Control Dynam* 29(6):1339–1349
22. Höhne G (2000) Computer aided development of biomechanical pilot models. *Aero Sci Tech* 4:57–69
23. Jagacinski RJ (2003) *Control theory for humans – Quantitative approaches to modeling performance*. Erlbaum, Mahwah, NJ
24. Kleinman DL, Baron S, Levison WH (1970) An optimal control model of human response. Part I: theory and validation. Part II: prediction of human performance in a complex task. *Automatica* 6:357–38

25. Kramer U (1985) On the application of fuzzy sets to the analysis of the system-driver-vehicle environment. *Automatica* 3(1):101–107
26. McRuer DT, Krendel ES (1974) Mathematical models of human pilot behavior. NATO AGAR-Dograph No. 188, Paris, France
27. Van Paassen R (1994) Biophysics in aircraft control: a model of the neuromuscular system of the pilot's arm. Ph.D. Thesis, Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands
28. Veeraklaew T, Agrawal SK (2001) New computational framework for trajectory optimization of higher-order dynamic systems. *J Guid Control Dynam* 24(2):228–236
29. Zeyada Y, Hess RA (2000) Modelling human pilot cue utilization with application to simulator fidelity assessment. *J Aircraft* 37(4):558–597



# Optimization of Multibody Systems and Their Structural Components

Olivier Brüls, Etienne Lemaire, Pierre Duysinx, and Peter Eberhard

**Abstract** This chapter addresses the optimization of flexible multibody systems based on the dynamic response of the full system with large amplitude motions and elastic deflections. The simulation model involves a nonlinear finite element formulation, a time integration scheme and a sensitivity analysis and it can be efficiently exploited in an optimization loop.

In particular, the paper focuses on the topology optimization of structural components embedded in multibody systems. Generally, topology optimization techniques consider that the structural component is isolated from the rest of the mechanism and use simplified quasi-static load cases to mimic the complex loadings in service. In contrast, we show that an optimization directly based on the dynamic response of the flexible multibody system leads to a more integrated approach.

The method is applied to truss structural components. Each truss is represented by a separate structural universe of beams with a topology design variable attached to each one. A SIMP model (or a variant of the power law) is used to penalize intermediate densities. The optimization formulation is stated as the minimization of the mean compliance over a time period or as the minimization of the mean tip deflection during a given trajectory, subject to a volume constraint. In order to illustrate the benefits of the integrated design approach, the case of a two degrees-of-freedom robot arm is developed.

## 1 Introduction

This chapter addresses the optimization of flexible multibody systems with large amplitude motions and elastic deflections. For example, in deployable space structures, piston engines, automotive suspensions, robots and high-speed machine-tools,

---

O. Brüls (✉), E. Lemaire, and P. Duysinx  
Department of Aerospace and Mechanical Engineering (LTAS), University of Liège, Belgium  
e-mail: [o.bruls@ulg.ac.be](mailto:o.bruls@ulg.ac.be); [e.lemaire@ulg.ac.be](mailto:e.lemaire@ulg.ac.be); [p.duysinx@ulg.ac.be](mailto:p.duysinx@ulg.ac.be)

P. Eberhard  
Institute of Engineering and Computational Mechanics, University of Stuttgart, Germany  
e-mail: [eberhard@itm.uni-stuttgart.de](mailto:eberhard@itm.uni-stuttgart.de)

the articulated components undergo large displacements and elastic deformations, and are subject to transient loads and nonlinear dynamic effects. The performance of such systems often depends on the mechanical design in a non-intuitive way.

Several researchers have addressed the optimization of the geometric parameters of rigid mechanisms, see among others [9, 18, 24], and also of the connectivity of mechanisms made of rigid members as in [27, 36]. Synthesis methods based on exhaustive search among possible combinations of links and joints were studied in [35]. Optimization techniques have also been exploited to solve optimal control problems in multibody dynamics, see for instance [11].

Initially developed in structural optimization, *topology optimization techniques* have often been used to optimize the layout of isolated linear elastic structural components under fixed loadings. Layout optimization of structures without any prior knowledge on the structural topology can be worked out by formulating the problem as an optimal material distribution on a given design domain, see [8] for details. As the optimal material distribution problem is generally solved numerically using a finite element discretization approach, the design domain is divided into finite elements and an existence variable is attached to each element. The optimal material distribution problem could be solved as a discrete valued problem, but this approach would require extensive computational resources because of its highly combinatorial nature. The solution of the discrete problem can be avoided by considering an alternative formulation in which the discrete existence variables are replaced by continuous density parameters running from void to solid via all intermediate densities. The density field may be interpreted as the spatial distribution of a fictitious porous material. This continuous formulation presents the advantage to allow using sensitivity analysis and mathematical programming algorithms to solve the problem in an efficient way.

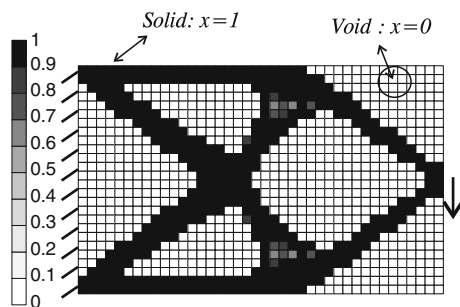
In most cases, the modelling of the intermediate density properties is based on the power-law model, also called SIMP model [6]. The effective Young's modulus  $E$  and the effective density  $\rho$  are given in term of the continuous existence variable  $x$  by

$$E = x^p E_0, \quad \rho = x \rho_0 \quad (1)$$

where the index 0 denotes the solid material properties. The factor  $p$  is introduced to penalize the intermediate densities in order to end up with contrasted “black and white” designs. A typical topology optimization result is presented in Fig. 1. In the final density map, black elements represent solid parts which belong to the optimal structure while white elements represent voids without any mechanical resistance.

Some extensions of topology optimization have been proposed for components with nonlinear geometric conditions [15, 34], nonlinear material behaviour [29] or fast dynamic effects [33]. In our particular case, interesting extensions of topology optimization are also concerned with the design of *compliant mechanisms* [32, 37]. In this case, the mechanism is considered as a whole, and the design results in massive beam-like components with compliant hinges. The compliant mechanisms treated in these works are usually not subject to high inertia effects coming from fast

**Fig. 1** Formulation of topology optimization as an optimal material distribution



motions of the motorized joints as in multibody systems dynamics and the compliant hinges generally do not undergo large rotations as in kinematic joints (hinges, sliders, etc.) under the action of attached motors or actuators.

This paper addresses the optimization of mechanisms composed of structural components and discrete kinematic joints. Due to their large motions, such mechanisms cannot be modelled as compliant mechanisms, but must be treated as *multibody systems*. When applying classical topology optimization techniques, one may consider that each structural component is isolated from the rest of the mechanism and use simplified quasi-static load cases to mimic the complex loadings in service. However, two main drawbacks are associated with this approach. Firstly, defining the equivalent load cases is a rather difficult task, which is often based on trials and errors and which requires some expertise. In [26], a systematic definition of the quasi-static loads from the transient response of the system is proposed. However, this method leads to a large number of load cases (typically, one load case per time step) and both the transient analysis and the static optimization procedure have to be repeated several times if the loads strongly depend on the mechanical design. Secondly, topology optimization is often sensitive to loading conditions, especially for multiple load cases and stress constraints (see [8] for illustrative examples), so that the optimal character of the resulting design can become questionable if the loading is approximative. For these reasons, in order to obtain better optimal layouts, this paper proposes an optimization procedure directly based on the dynamic response of the full flexible multibody system.

Literature reports some attempts to combine topology optimization with multibody dynamics for the design of structural components. In [2, 3, 31], a design procedure is proposed where each iteration of the optimization process involves two sequential steps. First, the dynamic response is computed using a model of the flexible multibody system, which is based on a floating frame of reference approach and on a modal representation of the structural flexibility. Second, the topology optimization is performed and the finite element model of the structural component is updated accordingly. At each iteration, the modal representation of the structural components should thus be recomputed from the updated finite element model. Considering the complexity of the resulting software architecture, it seems that the computation of the sensitivities is a challenging problem, so that gradient-based optimization techniques may not be utilized without significant approximations.

In order to overcome those limitations, a more integrated topology optimization technique is proposed here, based on the nonlinear finite element approach for flexible multibody systems described in [23]. The method is similar to the usual approach used in topology optimization in which the continuum domain is discretized into finite elements, see [8]. The nonlinear finite element formalism accounts for both large rigid-body motions and elastic deflections of the structural components. The design variables are classically density-like parameters associated to a SIMP law interpolation of effective material properties, see (1).

The nonlinear equations of motion are solved using a generalized- $\alpha$  time integration scheme, see for instance [4], and the sensitivity analysis of mechanical responses is based on a direct differentiation method as described in [12]. The efficient solution of the optimization problem relies on the sequential convex programming concept at the core of the CONLIN software [22].

In the present study, the method is applied to truss components, which are modelled using the flexible beam finite element available in our multibody simulation code. Each truss is represented by a structural universe of beams with a topology (i.e. existence) design variable attached to each one. The optimization formulation can be stated as the minimization of the mean compliance over a time period or as the minimization of the mean tip deflection during a given trajectory, subject to a volume constraint. In order to illustrate the benefits of the integrated design approach, the case of a two degrees-of-freedom robot arm is developed.

## 2 Optimization of Flexible Multibody Systems

### 2.1 Equations of Motion

A flexible multibody system can be modelled using the nonlinear finite element method proposed in [23]. After finite element discretization, the motion of each flexible body is represented by absolute nodal coordinates, which are gathered in the vector  $\mathbf{q}$ . The kinematic joints which connect the different bodies impose a set of nonlinear kinematic constraints between nodal coordinates, which are noted as  $\Phi(\mathbf{q}, t) = \mathbf{0}$ . The model of a multibody system has the general form

$$\mathbf{M}(\mathbf{q}, \mathbf{x})\ddot{\mathbf{q}} = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{x}, t) - \Phi_{,\mathbf{q}}^T \boldsymbol{\lambda}, \quad (2)$$

$$\Phi(\mathbf{q}, \mathbf{x}, t) = \mathbf{0} \quad (3)$$

with the initial conditions at time  $t = 0$  s

$$\mathbf{q}(0) = \mathbf{q}_0(\mathbf{x}), \quad (4)$$

$$\dot{\mathbf{q}}(0) = \dot{\mathbf{q}}_0(\mathbf{x}). \quad (5)$$

Equation (2) represents the dynamic equilibrium and (3) the kinematic constraints. The mass matrix is denoted by  $\mathbf{M}$ , which is not constant in case of large rotations,  $\mathbf{g} = \mathbf{g}^{ext} - \mathbf{g}^{int} - \mathbf{g}^{dam} - \mathbf{g}^{gyr}$  gathers the external, internal, damping and complementary inertia forces,  $\Phi_{,q}$  is the constraint gradient and  $\lambda$  is the vector of Lagrange multipliers.

The equations of motion (2) and (3) depend on a set of  $n$  design variables  $\mathbf{x}$  which can be related with the geometry of the system, its topology, its physical data or its applied loads. For given values of the design parameters  $\mathbf{x}$ , the dynamic response  $\mathbf{q}(\mathbf{x}, t)$ ,  $\lambda(\mathbf{x}, t)$  is defined as the solution to the system of differential-algebraic equations (2)–(5).

## 2.2 Formulation of the Optimization Problem

We consider the general form of optimization problems in multibody dynamics with inequality constraints and bounds for the design variables

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_0(\mathbf{x}) \\ \text{s.t.} \quad & \begin{cases} f_j(\mathbf{x}) \leq \bar{f}_j, & j = 1, \dots, m \\ \underline{x}_i \leq x_i \leq \bar{x}_i, & i = 1, \dots, n \end{cases} \end{aligned} \quad (6)$$

where the objective function  $f_0(\mathbf{x})$  and the design constraints  $f_j(\mathbf{x})$  ( $j \geq 1$ ) depend on the dynamic response  $\mathbf{q}(\mathbf{x}, t)$ ,  $\lambda(\mathbf{x}, t)$ . Introducing the compact notation

$$\mathbf{z}^T(\mathbf{x}, t) = [\mathbf{q}^T(\mathbf{x}, t) \quad \dot{\mathbf{q}}^T(\mathbf{x}, t) \quad \ddot{\mathbf{q}}^T(\mathbf{x}, t) \quad \lambda^T(\mathbf{x}, t)], \quad (7)$$

the objective function and the design constraints take the general form

$$\mathbf{f}(\mathbf{x}) = \int_0^{t_f} \mathbf{G}(\mathbf{z}(\mathbf{x}, t), \mathbf{x}, t) dt + \mathbf{F}^0(\mathbf{z}(\mathbf{x}, 0), \mathbf{x}) + \mathbf{F}^f(\mathbf{z}(\mathbf{x}, t_f), \mathbf{x}, t_f). \quad (8)$$

In this expression, the integrand  $\mathbf{G}$  accounts for the dynamic behaviour during the complete time interval  $[0, t_f]$ , while the functions  $\mathbf{F}^0$  and  $\mathbf{F}^f$  specifically account for the initial and final states. The following developments could be extended to situations where the final time  $t_f$  also depends on the design variables.

## 2.3 Time Integration Method

Equations (2) and (3) form a set of nonlinear differential and algebraic equations. As suggested in [23], it can be solved using the generalized- $\alpha$  method [17]. Despite the presence of algebraic constraints and despite the non-constant character of the mass

matrix, this scheme leads to accurate and reliable results, provided the introduction of a small amount of numerical damping [4]. Let us briefly describe the formulation of this algorithm.

At time step  $n + 1$ , the numerical variables  $\mathbf{q}_{n+1}$ ,  $\dot{\mathbf{q}}_{n+1}$ ,  $\ddot{\mathbf{q}}_{n+1}$  and  $\boldsymbol{\lambda}_{n+1}$  have to satisfy the coupled system (2) and (3). According to the generalized- $\alpha$  method, a vector  $\mathbf{a}$  of acceleration-like variables is defined by the recurrence relation

$$(1 - \alpha_m)\mathbf{a}_{n+1} + \alpha_m\mathbf{a}_n = (1 - \alpha_f)\ddot{\mathbf{q}}_{n+1} + \alpha_f\ddot{\mathbf{q}}_n, \quad \mathbf{a}_0 = \ddot{\mathbf{q}}_0, \quad (9)$$

and the integration scheme is obtained using  $\mathbf{a}$  in the Newmark integration formulae

$$\mathbf{q}_{n+1} = \mathbf{q}_n + h\dot{\mathbf{q}}_n + h^2 \left( \frac{1}{2} - \beta \right) \mathbf{a}_n + h^2 \beta \mathbf{a}_{n+1}, \quad (10)$$

$$\dot{\mathbf{q}}_{n+1} = \dot{\mathbf{q}}_n + h(1 - \gamma)\mathbf{a}_n + h\gamma\mathbf{a}_{n+1} \quad (11)$$

where  $h$  denotes the step size. Second-order accuracy and unconditional stability is guaranteed if the algorithmic parameters  $\alpha_f$ ,  $\alpha_m$ ,  $\beta$  and  $\gamma$  are properly selected according to [17].

For one time step, the numerical solution is computed using Algorithm 1, which actually solves (9)–(11) together with the dynamic equilibrium at time  $t_{n+1}$ . The Newton iterations try to bring the residuals  $\mathbf{r} = \mathbf{M}\ddot{\mathbf{q}} - \mathbf{g} + \boldsymbol{\Phi}_{,q}^T \boldsymbol{\lambda}$  and  $\boldsymbol{\Phi}$  to zero using the linearized form of (2) and (3)

$$\mathbf{M}\Delta\ddot{\mathbf{q}} + \mathbf{C}_t\Delta\dot{\mathbf{q}} + \mathbf{K}_t\Delta\mathbf{q} + \boldsymbol{\Phi}_{,q}^T\Delta\boldsymbol{\lambda} = \Delta\mathbf{r}, \quad (12)$$

$$\boldsymbol{\Phi}_{,q}\Delta\mathbf{q} = \Delta\boldsymbol{\Phi} \quad (13)$$

where  $\mathbf{C}_t = \partial\mathbf{r}/\partial\dot{\mathbf{q}}$  and  $\mathbf{K}_t = \partial\mathbf{r}/\partial\mathbf{q}$  denote the tangent damping and stiffness matrices. It can be demonstrated, see [4], that the iteration matrix of the algorithm is given by

$$\mathbf{S}_t = \begin{bmatrix} (\beta'\mathbf{M} + \gamma'\mathbf{C}_t + \mathbf{K}_t) & \boldsymbol{\Phi}_{,q}^T \\ \boldsymbol{\Phi}_{,q} & \mathbf{0} \end{bmatrix}$$

with  $\beta' = (1 - \alpha_m)/(h^2\beta(1 - \alpha_f))$  and  $\gamma' = \gamma/(h\beta)$ .

## 2.4 Evaluation of the Objective Function and of the Design Constraints

In order to evaluate the objective function and the design constraints  $\mathbf{f}(\mathbf{x})$ , it is convenient to introduce an intermediate variable  $\mathbf{y}(\mathbf{x}, t)$  defined by the differential equation

$$\dot{\mathbf{y}}(\mathbf{x}, t) = \mathbf{G}(\mathbf{z}(\mathbf{x}, t), \mathbf{x}, t) \quad (14)$$

---

**Algorithm 1**  $[\mathbf{q}_{n+1}, \dot{\mathbf{q}}_{n+1}, \ddot{\mathbf{q}}_{n+1}, \boldsymbol{\lambda}_{n+1}, \mathbf{a}_{n+1}] = \text{AlphaStep}(\mathbf{q}_n, \dot{\mathbf{q}}_n, \ddot{\mathbf{q}}_n, \mathbf{a}_n)$ 


---

```

 $\ddot{\mathbf{q}}_{n+1} := \mathbf{0}$ 
 $\mathbf{a}_{n+1} := 1/(1 - \alpha_m)(\alpha_f \ddot{\mathbf{q}}_n - \alpha_m \mathbf{a}_n)$ 
 $\mathbf{q}_{n+1} := \mathbf{q}_n + h\dot{\mathbf{q}}_n + h^2(0.5 - \beta)\mathbf{a}_n + h^2\beta\mathbf{a}_{n+1}$ 
 $\dot{\mathbf{q}}_{n+1} := \dot{\mathbf{q}}_n + h(1 - \gamma)\mathbf{a}_n + h\gamma\mathbf{a}_{n+1}$ 
 $\boldsymbol{\lambda}_{n+1} := \mathbf{0}$ 
for  $i = 1$  to  $i_{max}$  do
  Compute the residuals  $\mathbf{r}$  and  $\boldsymbol{\Phi}$ 
  if  $\sqrt{\|\mathbf{r}\|^2 + \|\boldsymbol{\Phi}\|^2} < tol$  then
    break
  end if
   $\begin{bmatrix} \Delta \mathbf{q} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} := -\mathbf{S}_t^{-1} \begin{bmatrix} \mathbf{r} \\ \boldsymbol{\Phi} \end{bmatrix}$ 
   $\mathbf{q}_{n+1} := \mathbf{q}_{n+1} + \Delta \mathbf{q}$ 
   $\dot{\mathbf{q}}_{n+1} := \dot{\mathbf{q}}_{n+1} + \gamma' \Delta \mathbf{q}$ 
   $\ddot{\mathbf{q}}_{n+1} := \ddot{\mathbf{q}}_{n+1} + \beta' \Delta \mathbf{q}$ 
   $\boldsymbol{\lambda}_{n+1} := \boldsymbol{\lambda}_{n+1} + \Delta \boldsymbol{\lambda}$ 
end for
 $\mathbf{a}_{n+1} := \mathbf{a}_{n+1} + (1 - \alpha_f)/(1 - \alpha_m)\ddot{\mathbf{q}}_{n+1}$ 

```

---

with the initial condition

$$\mathbf{y}(\mathbf{x}, 0) = \mathbf{F}^0(\mathbf{z}(\mathbf{x}, 0), \mathbf{x}). \quad (15)$$

As a consequence,  $\mathbf{f}(\mathbf{x})$  is computed from  $\mathbf{y}$  according to

$$\mathbf{f}(\mathbf{x}) = \mathbf{y}(\mathbf{x}, t_f) + \mathbf{F}^f(\mathbf{z}(\mathbf{x}, t_f), \mathbf{x}, t_f). \quad (16)$$

Equation (14) is easily solved by time integration, for instance, using an adaptation of Algorithm 1 for first-order differential equations [13]. Accordingly, an intermediate variable  $\mathbf{w}$  is introduced such that

$$(1 - \alpha_m)\mathbf{w}_{n+1} + \alpha_m\mathbf{w}_n = (1 - \alpha_f)\dot{\mathbf{y}}_{n+1} + \alpha_f\dot{\mathbf{y}}_n, \quad \mathbf{w}_0 = \dot{\mathbf{y}}_0, \quad (17)$$

and the integration scheme is obtained using  $\mathbf{w}$  in the integration formulae

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h(1 - \gamma)\mathbf{w}_n + h\gamma\mathbf{w}_{n+1}. \quad (18)$$

## 2.5 Sensitivity Analysis

Gradient-based optimization codes usually require the sensitivities of the simulation results with respect to the design parameters. For problems involving a rather large number of design parameters, e.g. in topology optimization, the efficient and reliable computation of those sensitivities is an important issue. Indeed, the finite difference technique, which is based on repeated simulations with perturbed values

of each design parameter, quickly becomes inefficient in this case. Therefore, more efficient algorithms should be implemented for the sensitivities, such as automatic differentiation [20] or semi-analytical approaches [10, 12].

In this paper, the semi-analytical direct differentiation method is presented for the computation of the sensitivities, i.e. the sensitivities are computed by differentiation of the time integration algorithm. For a given design parameter  $x$ , we use the notation  $(\bullet)' = \partial(\bullet)/\partial x$ . The sensitivity of the objective function and of the design constraints with respect to this design parameter is computed using the chain rule of differentiation

$$\mathbf{f}' = \mathbf{y}'(\mathbf{x}, t_f) + \mathbf{F}_{,z}^f(\mathbf{z}(\mathbf{x}, t_f), \mathbf{x}, t_f) \mathbf{z}'(\mathbf{x}, t_f) + \mathbf{F}_{,x}^f(\mathbf{z}(\mathbf{x}, t_f), \mathbf{x}, t_f) \quad (19)$$

where  $\mathbf{F}_{,z}^f$  (resp.  $\mathbf{F}_{,x}^f$ ) represents the partial derivative of  $\mathbf{F}^f$  with respect to the parameter  $\mathbf{z}$  (resp.  $x$ ). In this expression, the sensitivities  $\mathbf{z}'$  and  $\mathbf{y}'$  can be computed as described below.

According to the direct differentiation method, the sensitivities  $\mathbf{z}' = (\mathbf{q}', \dot{\mathbf{q}}', \ddot{\mathbf{q}}', \boldsymbol{\lambda}')$  are obtained by solving the differentiated form of (2) and (3)

$$\mathbf{M}\ddot{\mathbf{q}}' + \mathbf{C}_t\dot{\mathbf{q}}' + \mathbf{K}_t\mathbf{q}' + \boldsymbol{\Phi}_{,q}^T\boldsymbol{\lambda}' + \mathbf{r}_{,x} = \mathbf{0}, \quad (20)$$

$$\boldsymbol{\Phi}_{,q}\mathbf{q}' + \boldsymbol{\Phi}_{,x} = \mathbf{0} \quad (21)$$

with the initial conditions

$$\mathbf{q}'(0) = \mathbf{q}'_0, \quad (22)$$

$$\dot{\mathbf{q}}'(0) = \dot{\mathbf{q}}'_0. \quad (23)$$

The partial derivatives  $\mathbf{r}_{,x}$  and  $\boldsymbol{\Phi}_{,x}$  which appear in those equations are sometimes referred to as *pseudo-loads*. Even though the dynamic equilibrium is nonlinear with respect to  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ ,  $\ddot{\mathbf{q}}$  and  $\boldsymbol{\lambda}$ , one observes that the sensitivity equations (20) and (21) are linear with respect to  $\mathbf{q}'$ ,  $\dot{\mathbf{q}}'$ ,  $\ddot{\mathbf{q}}'$  and  $\boldsymbol{\lambda}'$ .

At time step  $n + 1$ , the sensitivities can be computed using the same integration algorithm as for the dynamic response, i.e.

$$[\mathbf{q}'_{n+1}, \dot{\mathbf{q}}'_{n+1}, \ddot{\mathbf{q}}'_{n+1}, \boldsymbol{\lambda}'_{n+1}, \mathbf{a}'_{n+1}] = \text{AlphaStep}'(\mathbf{q}'_n, \dot{\mathbf{q}}'_n, \ddot{\mathbf{q}}'_n, \mathbf{a}'_n). \quad (24)$$

More precisely, *AlphaStep'* is the same algorithm as *AlphaStep*, excepted that the residuals  $\mathbf{r}$  and  $\boldsymbol{\Phi}$  are replaced by the residuals of (20) and (21). Since (20) and (21) are linear, a single Newton iteration is sufficient to get the exact values of the sensitivities at the current time step. In particular, the iteration matrix  $\mathbf{S}_t$  is the same as for the original problem. Hence, even for a large number of design parameters, this matrix should be computed and factorized only once for the sensitivity analysis at time step  $n + 1$ . However, while an *approximate* iteration matrix is often sufficient to achieve convergence for the original problem, an *exact* expression is necessary to solve the sensitivity problem in a single iteration.

In a similar way, the sensitivity  $\mathbf{y}'$  satisfies the differentiated form of (14)

$$\dot{\mathbf{y}}' = \mathbf{G}_{,z}\mathbf{z}' + \mathbf{G}_{,x} \quad (25)$$



**Algorithm 2**  $[\mathbf{f}, \mathbf{f}'] = \text{ObjectiveFunctionAndSensitivities}(x)$ 


---

```

initialize  $\mathbf{q}_0, \dot{\mathbf{q}}_0, \ddot{\mathbf{q}}_0, \mathbf{a}_0, \mathbf{y}_0, \dot{\mathbf{y}}_0, \mathbf{w}_0$ 
initialize  $\mathbf{q}'_0, \dot{\mathbf{q}}'_0, \ddot{\mathbf{q}}'_0, \mathbf{a}'_0, \mathbf{y}'_0, \dot{\mathbf{y}}'_0, \mathbf{w}'_0$ 
for  $n = 0$  to  $n_f - 1$  do
     $[\mathbf{q}_{n+1}, \dot{\mathbf{q}}_{n+1}, \ddot{\mathbf{q}}_{n+1}, \lambda_{n+1}, \mathbf{a}_{n+1}] = \text{AlphaStep}(\mathbf{q}_n, \dot{\mathbf{q}}_n, \ddot{\mathbf{q}}_n, \mathbf{a}_n)$ 
     $\dot{\mathbf{y}}_{n+1} = \mathbf{G}(\mathbf{z}_{n+1}, \mathbf{x}, t_{n+1})$ 
     $\mathbf{w}_{n+1} = ((1 - \alpha_f)\dot{\mathbf{y}}_{n+1} + \alpha_f\dot{\mathbf{y}}_n - \alpha_m\mathbf{w}_n)/(1 - \alpha_m)$ 
     $\mathbf{y}_{n+1} = \mathbf{y}_n + h(1 - \gamma)\mathbf{w}_n + h\gamma\mathbf{w}_{n+1}$ 
     $[\mathbf{q}'_{n+1}, \dot{\mathbf{q}}'_{n+1}, \ddot{\mathbf{q}}'_{n+1}, \lambda'_{n+1}, \mathbf{a}'_{n+1}] = \text{AlphaStep}'(\mathbf{q}'_n, \dot{\mathbf{q}}'_n, \ddot{\mathbf{q}}'_n, \mathbf{a}'_n)$ 
     $\dot{\mathbf{y}}'_{n+1} = \mathbf{G}_{,x}\mathbf{z}' + \mathbf{G}_{,x}$ 
     $\mathbf{w}'_{n+1} = ((1 - \alpha_f)\dot{\mathbf{y}}'_{n+1} + \alpha_f\dot{\mathbf{y}}'_n - \alpha_m\mathbf{w}'_n)/(1 - \alpha_m)$ 
     $\mathbf{y}'_{n+1} = \mathbf{y}'_n + h(1 - \gamma)\mathbf{w}'_n + h\gamma\mathbf{w}'_{n+1}$ 
end for
 $\mathbf{f} = \mathbf{y}_{n_f} + \mathbf{F}^f(\mathbf{z}_{n_f}, \mathbf{x}, t_f)$ 
 $\mathbf{f}' = \mathbf{y}'_{n_f} + \mathbf{F}^f_{,x}(\mathbf{z}_{n_f}, \mathbf{x}, t_f)\mathbf{z}'_{n_f} + \mathbf{F}^f_{,x}(\mathbf{z}_{n_f}, \mathbf{x}, t_f)$ 

```

---

with the initial condition

$$\mathbf{y}'(0) = \mathbf{F}^0_{,q}\mathbf{q}'_0 + \mathbf{F}^0_{,\dot{q}}\dot{\mathbf{q}}'_0 + \mathbf{F}^0_{,x}. \quad (26)$$

In summary, the dynamic response, the objective function, the design constraints and their sensitivities can be computed efficiently in a single but extended simulation. In the case of a single design parameter, the complete procedure is described in Algorithm 2. The implementation effort to develop this algorithm in an existing simulation software is limited since the core routine *AlphaStep* does not need to be modified and since the sensitivity routine *AlphaStep'* closely resembles it.

## 2.6 Optimization Algorithms

Several optimization methods have been applied to solve problems in structural and applied mechanics. Depending on the problem characteristics and the available information, one or several of these methods can be selected. On the one hand, *heuristic methods* such as *Genetic Algorithms* [5] or *Particle Swarm methods* [36] are algorithms inspired by natural phenomena. These algorithms only require the computation of the design function values and their global convergence can be guaranteed even for non-convex problems with local minima. They can tackle problems with discrete valued variables and non-smooth functions. However, the main drawback of these methods is their slow convergence rate and the large number of function evaluations usually needed to reach the optimum, which results in a high computational load for large scale systems. Thus, they are generally restricted to problems with a small number of design variables (about 10). On the other hand, one can take advantage of *mathematical programming* (MP) methods, which usually require the design function derivatives. Their convergence speed is generally higher than for heuristic methods. MP methods have been successfully applied to solve

large scale structural and multidisciplinary optimization problems. Furthermore, due to their high convergence speed, the optimal solution can be obtained within a limited number of iterations and function evaluations. Amongst the well-known applications of mathematical programming, we can mention *CONLIN* [22], *MMA* and its extensions [14, 38, 39], *FAIPA* [25] or *IPOPT* [40].

CONLIN, MMA and its extensions are based on the so-called *sequential convex programming* approach, which relies on two concepts sketched in Fig. 2.

1. The original optimization problem in (6) that is highly non-linear and implicit in the design variables is replaced by a sequence of explicit and convex sub-problems, that are built based on a local approximation of design functions

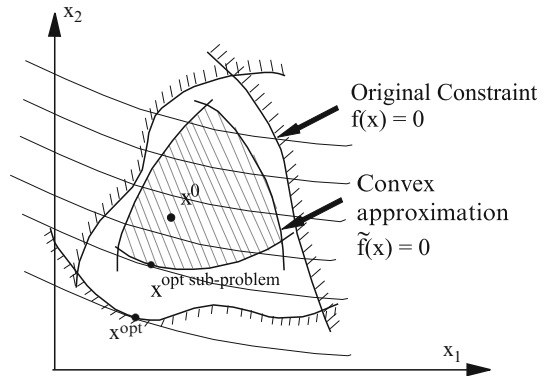
$$\begin{aligned} \min_{\mathbf{x}} \quad & \tilde{f}_0(\mathbf{x}) \\ \text{s.t.} \quad & \begin{cases} \tilde{f}_j(\mathbf{x}) \leq \bar{f}_j, & j = 1, \dots, m, \\ \underline{x}_i \leq x_i \leq \bar{x}_i, & i = 1, \dots, n. \end{cases} \end{aligned} \quad (27)$$

The local approximations are established using the sensitivities as well as a variant of the Taylor series expansion of the design functions.

2. Each local convex subproblem is solved efficiently using fast and effective mathematical programming algorithms such as Lagrangian maximization (dual method) or interior point methods.

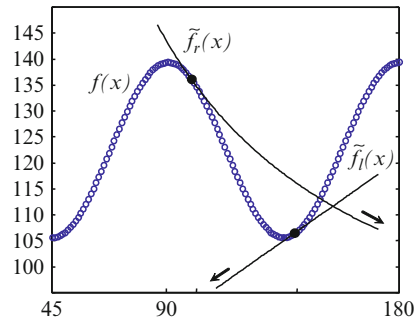
Dual methods allow to reach the optimum of the local convex subproblem within a limited number of iterations independent of the number of design variables. The concept has proved to be very general and efficient in topology optimization problems, see e.g. [15, 33, 37].

In the present work, CONLIN [22] has been selected for its fast convergence properties for large scale topology optimization problems. The sensitivities shall be efficiently evaluated as shown in Sect. 2. CONLIN, which is an acronym for CONvex LINearization, relies on a particular first order Taylor expansion



**Fig. 2** Iterative solution of structural optimization problems using a sequential convex programming approach

**Fig. 3** CONLIN approximation [14]



based on a combination of both direct variables and reciprocal variables  $1/x_i$ . The direct variable expansion is used when the first order derivative is positive while the reciprocal expansion is exploited when their first order derivative is negative

$$\tilde{f}_j(\mathbf{x}) = f_j(\mathbf{x}^0) + \sum_{+} \frac{\partial f_j}{\partial x_i} (x_i - x_i^0) - \sum_{-} (x_i^0)^2 \frac{\partial f_j}{\partial x_i} \left( \frac{1}{x_i} - \frac{1}{x_i^0} \right) \quad (28)$$

where  $\mathbf{x}^0$  is the expansion point in state space,  $\sum_{+}$  is the sum over all the terms for which the derivative is positive and  $\sum_{-}$  is the sum over all the terms for which the derivative is negative. As illustrated in Fig. 3,  $f(x)$  is approximated by a linear function  $\tilde{f}_l(x)$  or by a reciprocal function  $\tilde{f}_r(x)$  depending on the sign of its first derivative at the point of approximation. It can be demonstrated that the CONLIN scheme (28) is unconditionally convex and that it is the most conservative approximation that can be generated with linear and reciprocal variables. This means that the approximation (28) tends to lie in the feasible domain of the constraint. It follows that the CONLIN method mostly tends to generate feasible new solutions. The convex and separable character of the approximated functions allows the use of dual optimizers and second order maximization algorithms for the sub-problems.

For the applications treated in this paper, a coupled software interface is used where the dynamic response and the sensitivity analysis are computed using the OOFELIE finite element software [16] and the optimization of the design variables is achieved by the CONLIN software [22].

### 3 Topology Optimization Techniques

Topology optimization techniques rely on a finite element discretization of the continuous elastic domain. The design variables are the pseudo-densities (existence variables) of the elements of the mesh. The modelling of material properties with intermediate densities is based on the SIMP model in (1). This power law decreases the stiffness (i.e. the efficiency) of intermediate densities while the cost in terms of material volume stays linear. The available amount of material being limited, the use

of intermediate densities is penalized and the optimal design usually ends up with mostly void and solid regions. The exponent  $p$  in (1) is classically chosen equal to 3.

In the equations of motion, the density variables only appear in the expression of the inertia and internal elastic forces. Those forces are computed by a finite element assembly procedure

$$\mathbf{g}^{inert}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{x}) = \sum_{e=1}^{n_e} \mathbf{L}_e^T \mathbf{g}_e^{inert}(\mathbf{q}_e, \dot{\mathbf{q}}_e, \ddot{\mathbf{q}}_e, x_e), \quad (29)$$

$$\mathbf{g}^{int}(\mathbf{q}, \mathbf{x}) = \sum_{e=1}^{n_e} \mathbf{L}_e^T \mathbf{g}_e^{int}(\mathbf{q}_e, x_e) \quad (30)$$

where  $n_e$  is the number of elements of the mesh,  $\mathbf{g}^{inert}$  is the vector of inertia forces

$$\mathbf{g}^{inert}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{x}) = \mathbf{M}(\mathbf{q}, \mathbf{x})\ddot{\mathbf{q}} + \mathbf{g}^{gyr}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{x}) \quad (31)$$

and  $\mathbf{L}_e$  is the Boolean localization matrix of the element dofs

$$\mathbf{q}_e = \mathbf{L}_e \mathbf{q}. \quad (32)$$

Since the elementary inertia forces depend linearly on  $\rho$  and the elementary elastic forces depend linearly on  $E$ , the SIMP law yields

$$\mathbf{g}_e^{inert}(\mathbf{q}_e, \dot{\mathbf{q}}_e, \ddot{\mathbf{q}}_e, x_e) = x_e \mathbf{g}_e^{inert}(\mathbf{q}_e, \dot{\mathbf{q}}_e, \ddot{\mathbf{q}}_e, 1), \quad (33)$$

$$\mathbf{g}_e^{int}(\mathbf{q}_e, x_e) = x_e^p \mathbf{g}_e^{int}(\mathbf{q}_e, 1). \quad (34)$$

In those equations, the right-hand-side depends on the inertia force and the internal force of the full-density element, which is readily available in a standard finite element simulation software. As a consequence, the relation between the residual  $\mathbf{r}$  and the set of design parameters  $\mathbf{x}$  is known analytically and the pseudo-load can be computed on the element level

$$(\mathbf{r}, x)_e = \mathbf{g}_e^{inert}(\mathbf{q}_e, \dot{\mathbf{q}}_e, \ddot{\mathbf{q}}_e, 1) + p x_e^{p-1} \mathbf{g}_e^{int}(\mathbf{q}_e, 1). \quad (35)$$

The global pseudo-load is then obtained by numerical assembly according to

$$\mathbf{r}, x = \sum_{e=1}^{n_e} \mathbf{L}_e^T (\mathbf{r}, x)_e. \quad (36)$$

### 3.1 Application to the Design of Static Trusses

In order to illustrate the principle of topology optimization techniques, the design of static trusses is first considered. Truss topology optimization has been initially investigated by Michell [30] at the beginning of the 20th century. The application of

numerical methods to the truss optimization problem is more recent and has been proposed in [19] and [21] leading to the so-called ground structure approach, see [28] for a review. While the problem was classically formulated in terms of stresses, several developments have been performed to establish a displacement formulation of the problem, see e.g. [1, 7].

Starting from an initial truss, i.e. the ground structure, the objective of the method is to remove the less efficient structural members. In practice, the ground structure is often created by connecting a set of chosen nodes with rods or beams in all possible ways. Then, as for topology optimization of continuum structures, a density variable  $x$  is attached to each structural member. Inspired by topology optimization methods, the modelling of material properties with intermediate densities can be based on the SIMP model in (1).

For example, the design problem can often be stated as the minimization of the compliance

$$c = \frac{1}{2} \int_V \boldsymbol{\epsilon}^T \mathbf{H} \boldsymbol{\epsilon} dV \quad (37)$$

where  $\boldsymbol{\epsilon}$  denotes the strain vector,  $\mathbf{H}$  is the Hooke tensor, and  $V$  is the volume of the structure. A design constraint is imposed on the maximal volume of material

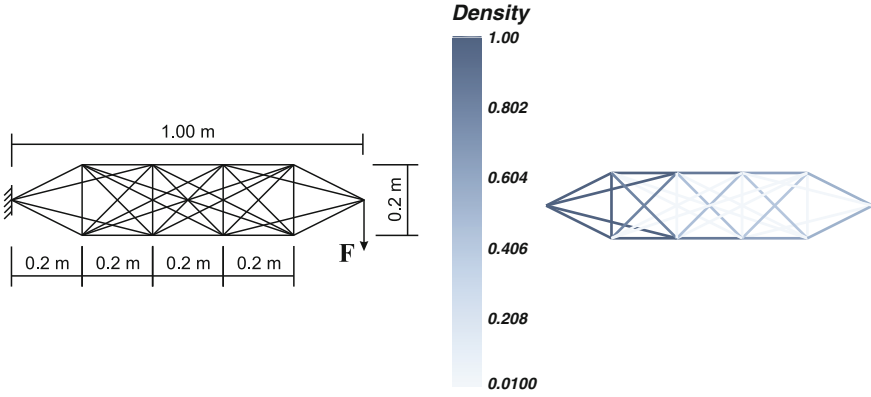
$$V = \sum_{e=1}^{n_e} x_e V_{full,e} \leq \eta V_{full} \quad (38)$$

where  $n_e$  is the number of bars,  $V_{full,e}$  is the volume of the full bar  $e$ ,  $V_{full}$  is the volume of the full truss and  $\eta$  is a coefficient between 0 and 1. Moreover, each density is bounded by  $x_{min} \leq x \leq 1$ . In the numerical applications presented hereafter, we have selected the values  $\eta = 0.4$  and  $x_{min} = 0.01$ .

In order to illustrate the topology optimization of static trusses, let us consider a linkage composed of cylindrical beams and subject to a static tip load, as shown in Fig. 4. The beams are made of aluminum (density = 2700 kg/m<sup>3</sup>, Young's modulus = 70E9 Pa, Poisson ratio = 0.32), and their diameter is 0.02 m. The minimization of the compliance (37) under the volume constraint (38) leads to the optimal truss represented in Fig. 4. After a few iterations, the density variables are either close to 0 or significantly larger than zero. It is noticeable that the optimal design, which results from a fully automatic optimization procedure, is acceptable for the engineering common sense.

## 3.2 Topology Optimization of Multibody Systems

A more challenging problem is the optimization of truss linkages included in multibody systems. In a classical approach, one would reformulate the dynamic problem as a set of static problems. In a first step, a rigid multibody software is used in order to precompute the loads applied to each component, and in a second step,



**Fig. 4** Initial and optimal truss for a static tip load

the topology of each linkage is optimized independently using the static approach described in the previous section. For this purpose, a set of static load cases should be defined in order to mimic the precomputed dynamic loads.

Alternatively, an integrated optimization approach is proposed here, which is based on a dynamic multibody analysis. The advantages are that: (i) the dynamic coupling between large overall rigid-body motions and deformations is properly taken into account, (ii) a single dynamic analysis is required by the optimizer instead of a patchwork of static analyses, (iii) topology-dependent loads can be considered, and (iv) the objective function and the design constraints may be defined with respect to the actual dynamic problem.

As seen in the previous section, static topology optimization problems are often formulated as the minimization of the compliance of the deformed structure. In order to extend this approach to dynamic problems, the objective function may be defined as the mean compliance over the trajectory

$$\bar{c} = \frac{1}{t_f} \int_0^{t_f} \sum_{i=1}^{n_c} c_{(i)} dt \quad (39)$$

where  $n_c$  represents the number of structural components and  $c_{(i)}$  is the instantaneous compliance of component  $i$  defined by (37).

For a robot arm, a task-oriented objective function could be preferred. For example, the mean squared tip-deflection over the trajectory is defined by

$$\bar{d} = \frac{1}{t_f} \int_0^{t_f} \|\mathbf{r} - \mathbf{r}_{rigid}\|^2 dt \quad (40)$$

where the vector  $\mathbf{r}$  represents the actual tip position and  $\mathbf{r}_{rigid}$  is the tip position of the undeformed mechanism. Clearly, a reduction of  $\bar{d}$  corresponds to an improvement in the trajectory tracking performances.

## 4 Example

### 4.1 Problem Description

Figures 5 and 6 represent a robot arm composed of two truss linkages interconnected by revolute joints and moving in a horizontal plane, so that gravity can be ignored. The parameters of each truss structure were given in Sect. 3.1. Moreover, a non-design mass of 5 kg is fixed at the tip of the manipulator.

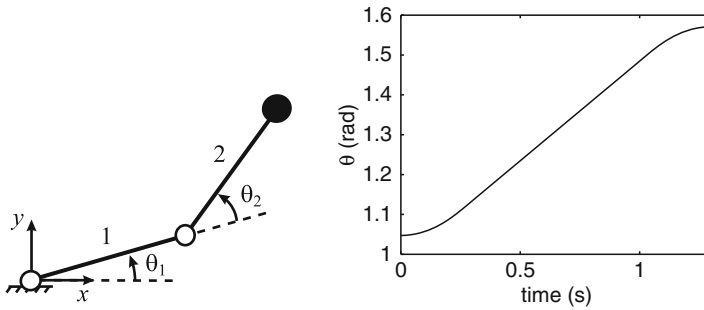


Fig. 5 Kinematic model. Imposed joint trajectory for  $\theta_1$  and  $\theta_2$

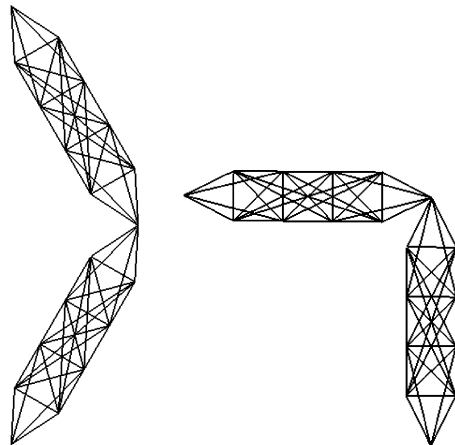


Fig. 6 Initial and final configurations

Each revolute joint is driven by an ideal motor which imposes a smooth joint trajectory  $\theta_1(t)$  and  $\theta_2(t)$ . As illustrated in Figs. 5 and 6, a point-to-point trajectory has been selected, which is composed of an acceleration phase, a constant speed phase, and a deceleration phase. The numerical data are  $\theta_{1i} = \theta_{2i} = \pi/3$  rad,  $\theta_{1f} = \theta_{2f} = \pi/2$  rad, acceleration (deceleration) = 2 rad/s<sup>2</sup>, velocity = 0.5 rad/s.

For each linkage separately, an upper bound is imposed to the volume of material

$$V_{(i)} \leq 0.4 V_{full,(i)}, \quad i = 1, 2 \quad (41)$$

where  $V_{(i)}$  is the volume of linkage  $i$ , see (38), and  $V_{full,(i)}$  is the volume of the linkage with full densities.

The simulation and the sensitivity analysis are realized using the numerical algorithm described in Sect. 2 with a time step  $h = 0.02$  s. The algorithmic coefficients  $\beta$ ,  $\gamma$ ,  $\alpha_f$  and  $\alpha_m$  are defined according to the method of Chung and Hulbert [17] and the value 0.8 is selected for the spectral radius at infinite frequencies.

## 4.2 Optimization

The minimization of the mean compliance defined by (39) leads to the design represented in Fig. 7. We observe that the mass densities are reduced as one progresses from the ground to the tip, i.e. that densities are lower in areas with large amplitude motions. A possible explanation is that the addition of materials in those areas increases the total amount of kinetic energy in the system, which leads to higher inertia forces and vibration excitations. We intend to continue our investigations in order to verify this interpretation and to check if this phenomenon is caused by the existence

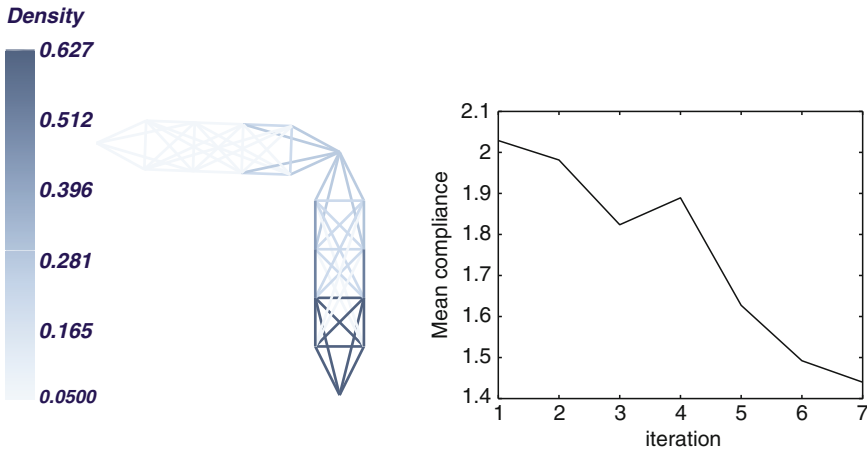


Fig. 7 Minimization of the mean compliance



of a local minimum. After seven iterations, the dynamic problem becomes badly conditioned and the time integrator is no more able to compute the response of the system.

Since the minimization of the mean compliance does not lead to an acceptable design, we have considered the minimization of the mean squared tip-deflection defined by (40). Actually, this formulation in terms of trajectory tracking performances is closer to the user expectations, which represents a clear advantage. The rigid trajectory  $\mathbf{r}_{rigid}$  is computed from the joint trajectory using the rigid kinematic relations of the manipulator. The resulting optimal design, which is represented in Fig. 8, is very attractive from an engineering point of view. Its superiority is further demonstrated in Fig. 9, which compares the tip deflections for the initial and optimized designs. One also observes that the final design is not the same for the two structural components, which makes sense since they are subject to different load

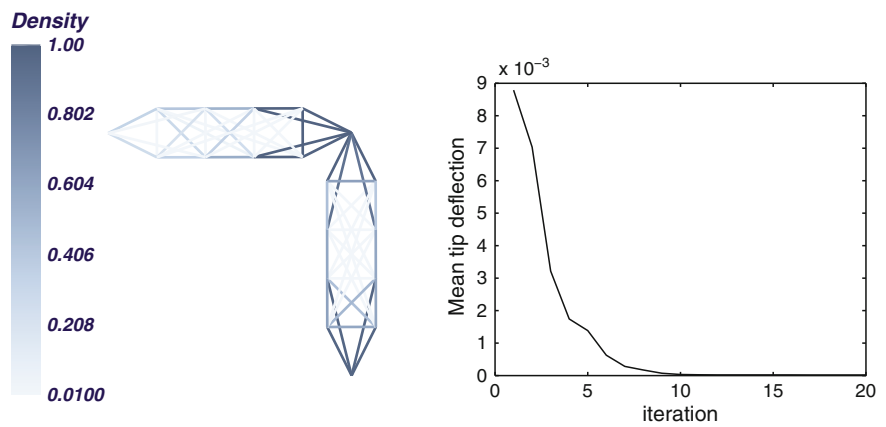


Fig. 8 Minimization of the tip-deflection

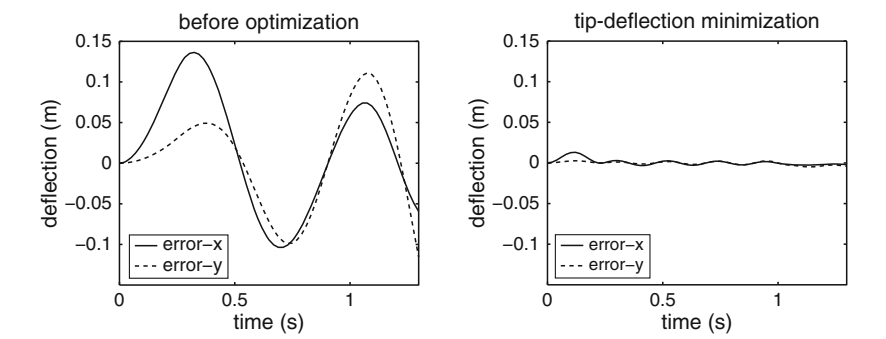


Fig. 9 Tip deflections

conditions. This again motivates the interest in the integrated optimization approach proposed in this paper, which properly accounts for the dynamic loads during the motion of the multibody system.

## 5 Conclusions

This chapter is about the optimization of flexible multibody systems based on their dynamic response. The proposed approach involves a nonlinear finite element formulation, a generalized- $\alpha$  time integrator, a direct differentiation sensitivity analysis and an optimization algorithm.

In particular, the topology optimization of linkages included in multibody systems is addressed. The material properties are described by a SIMP model and a sequential convex programming optimizer is used. The developments have been implemented in the OOFELIE simulation package, which is coupled with the CONLIN optimizer.

The methodology has been validated for the design of a two degrees-of-freedom robot arm with truss linkages. Due to important inertia effects, the classical compliance objective function does not yield a satisfactory design. In contrast, a task-oriented criterion, such as the mean tip deflection, appears to be a sound basis for the optimal design of the two links of the mechanism.

We conclude that optimization algorithms can rely efficiently on simulation tools for flexible multibody systems provided that algorithms for sensitivity analysis are implemented in the numerical time integration procedure. In particular, the integrated approach proposed in this work opens some perspectives for the development of topology optimization techniques for elastic components of mechanisms. Promising 3D applications can be found in various fields, such as deployable space structures, piston engines, automotive suspensions or high-speed robots and machine-tools.

## References

1. Achtziger W, Bendsøe M, Ben-Tal A, Zowe J (1992) Equivalent displacement based formulations for maximum strength truss topology design. *Impact Comput Sci Eng* 4(4):315–345
2. Albers A, Häussler P (2005) Topology optimization of dynamic loaded parts using multibody simulation and durability analysis. In: NAFEMS seminar ‘Optimization in structural mechanics.’ Wiesbaden, Germany
3. Albers A, Otténad J, Häussler P (2007) Structural optimization of components in controlled mechanical systems. In: *Proceedings of 6th International Conference on Multibody Systems, Nonlinear Dynamics, and Control*, ASME IDETC/CIE. Las Vegas, Nevada, USA
4. Arnold M, Brüls O (2007) Convergence of the generalized- $\alpha$  scheme for constrained mechanical systems. *Multibody Syst Dyn* 18(2):185–202
5. Arora J (2004) *Introduction to optimum design*, 2nd edn. Elsevier Academic Press, San Diego
6. Bendsøe M (1989) Optimal shape design as a material distribution problem. *Struct Optimization* 1:193–202

7. Bendsøe M, Ben-Tal A, Zowe J (1994) Optimization methods for truss geometry and topology design. *Struct Optimization* 7(3):141–158
8. Bendsøe M, Sigmund O (2003) *Topology optimization: theory, methods, and applications*. Springer Verlag, Berlin
9. Bestle D, Eberhard P (1992) Analyzing and optimizing multibody systems. *Mech Struct Mach* 20:67–92
10. Bestle D, Seybold J (1992) Sensitivity analysis of constrained multibody systems. *Arch Appl Mech* 62:181–190
11. Bottasso C, Croce A, Ghezzi L, Faure P (2004) On the solution of inverse dynamics and trajectory optimization problems for multibody systems. *Multibody Syst Dyn* 11:1–22
12. Brüls O, Eberhard P (2008) Sensitivity analysis for dynamic mechanical systems with finite rotations. *Int J Numer Meth Eng* 74(13):1897–1927
13. Brüls O, Golinval JC (2006) The generalized- $\alpha$  method in mechatronic applications. *J Appl Math Mech (ZAMM)* 86(10):748–758
14. Bruyneel M, Duysinx P, Fleury C (2004) A family of MMA approximations for structural optimization. *Struct Multidiscip O* 24(4):263–276
15. Buhl T, Pedersen C, Sigmund O (2000) Stiffness design of geometrically non-linear structures using topology optimization. *Struct Multidiscip O* 19(2):93–104
16. Cardona A, Klapka I, Gérardin M (1994) Design of a new finite element programming environment. *Eng Computation* 11:365–381
17. Chung J, Hulbert G (1993) A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized- $\alpha$  method. *ASME J Appl Mech* 60:371–375
18. Collard JF, Fiset P, Duysinx P (2005) Contribution to the optimization of closed-loop multibody systems: Application to parallel manipulators. *Multibody Syst Dyn* 13:69–84
19. Dorn W, Gomory R, Greenberg H (1964) Automatic design of optimal structures. *Jo de Mécanique* 3:25–52
20. Eberhard P, Bischof C (1999) Automatic differentiation of numerical integration algorithms. *Math Comput* 68:717–731
21. Fleron P (1964) The minimum weight of trusses. *Byggningsstatiske Meddelelser* 35:81–96
22. Fleury C (1989) CONLIN: an efficient dual optimizer based on convex approximation concepts. *Struct Optimization* 1(2):81–89
23. Gérardin M, Cardona A (2001) *Flexible multibody dynamics: A finite element approach*. Wiley, New York
24. Hansen J (2002) Synthesis of mechanisms using time-varying dimensions. *Multibody Syst Dyn* 7:127–144
25. Herskovits J, Mappa P, Goulart E, Soares C (2005) Mathematical programming models and algorithms for engineering design optimization. *Comput Meth Appl Mech Eng* 194(30–33):3244–3268
26. Kang B, Park G, Arora J (2005) Optimization of flexible multibody dynamic systems using the equivalent static load method. *AIAA J* 43(4):846–852
27. Kawamoto A, Bendsøe M, Sigmund O (2004) Articulated mechanism design with a degree of freedom constraint. *Int J Numer Meth Eng* 61(9):1520–1545
28. Kirsch U (1989) Optimal topologies of truss structures. *Comput Meth Appl Mech Eng* 72:15–28
29. Maute K, Schwarz S, Ramm E (1998) Adaptive topology optimization of elastoplastic structures. *Struct Optimization* 15(2):81–91
30. Michell A (1904) The limit of economy of material in frame structures. *Phil Mag* 8(6):589–597
31. Müller O, Häussler P, Lux R, Ilzhöfer B, Albers A (1999) Automated coupling of MDI/ADAMS and MSC.CONSTRUCT for the topology and shape optimization of flexible mechanical systems. In: *ADAMS users conference*. Berlin, Germany
32. Nishiwaki S, Frecker M, Min S, Kikuchi N (1998) Topology optimization of compliant mechanisms using the homogenization method. *Int J Numer Meth Eng* 42(3):535–559
33. Pedersen C (2003) Topology optimization for crashworthiness of frame structures. *Int J Crashworthiness* 8(1):29–39

34. Pedersen C, Buhl T, Sigmund O (2001) Topology synthesis of large-displacement compliant mechanisms. *Int J Numer Meth Eng* 50:2683–2705
35. Pucheta M, Cardona A (2007) An automated method for type synthesis of planar linkages based on a constrained subgraph isomorphism detection. *Multibody Syst Dyn* 18(2):233–258
36. Sedlaczek K, Eberhard P (2007) Augmented Lagrangian particle swarm optimization in mechanism design. *J Sys Design Dyn* 1(3):410–421
37. Sigmund O (1997) On the design of compliant mechanisms using topology optimization. *Mech Struct Mach* 25:493–524
38. Svanberg K (1987) The method of moving asymptotes – a new method for structural optimization. *Int J Numer Meth Eng* 24:359–373
39. Svanberg K (2002) A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM J Optim* 12(2):555–573
40. Waechter A, Biegler LT (2006) On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Math Program* 106(1):25–57

# Real-Time Aeroservoelastic Analysis of Wind-Turbines by Free Multibody Software

Luca Cavagna, Alessandro Fumagalli, Pierangelo Masarati,  
Marco Morandini, and Paolo Mantegazza

**Abstract** This work illustrates the feasibility of the implementation of innovative, efficient and low-cost solutions for fast-prototyping and customization of controlled mechanical and aeroservoelastic systems. A controlled constant-speed wind-turbine is considered as an example of the proposed methodologies, where the physical aeromechanics problem is controlled by a controller process scheduled for execution in real-time on a PC-class computer. The physical problem is simulated by a general-purpose multibody process that is scheduled in real-time as well. The processes communicate using real-time inter-process communication primitives. All the involved tools are based on free software.

## 1 Introduction

Wind-turbines represent an important means to extract energy from the environment in a ‘green’ manner. The concept of extracting energy from the wind dates back thousands of years, including not only power generation (e.g. mills, water pumps) but also direct locomotion (e.g. sailing). Modern wind-energy technology relies on efficient aerodynamic design and durable mechanical systems.

Nonetheless, efficient and reliable energy harvesting from winds poses several significant challenges, including:

- optimal harvesting with rather irregular and erratic wind conditions
- tolerance to wear and fatigue with minimal maintenance

---

L. Cavagna (✉)  
FOI (Swedish Defence Research Agency), Sweden  
e-mail: [luca.cavagna@polimi.it](mailto:luca.cavagna@polimi.it)

A. Fumagalli, P. Masarati, M. Morandini, and P. Mantegazza  
Politecnico di Milano, Dipartimento di Ingegneria Aerospaziale, Italy  
e-mail: [alessandro.fumagalli@polimi.it](mailto:alessandro.fumagalli@polimi.it); [pierangelo.masarati@polimi.it](mailto:pierangelo.masarati@polimi.it);  
[marco.morandini@polimi.it](mailto:marco.morandini@polimi.it); [paolo.mantegazza@polimi.it](mailto:paolo.mantegazza@polimi.it)

- controllability and survivability during exceptional weather and operating conditions
- efficient and fault-tolerant integration with power grids

It is anticipated that key to many of the issues mentioned above is control. A recent, extensive review of the state of the art in wind-turbines aerodynamics and aeroelasticity is presented in [1]. Smart rotor control research for wind-turbines is presented in [2].

The goal of this work is to illustrate the rapid feasibility of the implementation of innovative, efficient and low-cost solutions for fast-prototyping and customization of controlled mechanical and aeroservoelastic systems. The specific problem of designing control systems for wind turbines is not addressed. The work rather focuses on providing analysis tools that can be used for this purpose.

## 2 Approach

The free Real-Time Operating System (RTOS) Real-Time Application Interface (RTAI [3]) and the free general-purpose multibody software MBDyn [4], both originating from research at the Dipartimento di Ingegneria Aerospaziale of Politecnico di Milano, Italy, are at the core of the present work. The use of free software that runs on low-cost hardware gives any organization, significantly the academia and Small-Medium Enterprises (SMEs), access to powerful and versatile analysis and simulation capabilities.

A controlled constant-speed wind-turbine [5] is considered as one of the possible applications of the developed methodologies. Figure 1 shows a sketch of

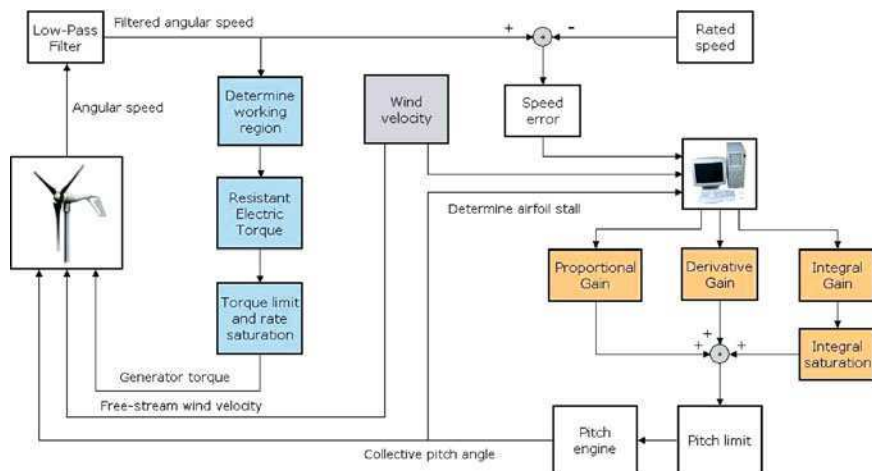


Fig. 1 Sketch of the controlled model

the controlled model under analysis [6]. It consists in a multibody model of the Controls Advanced Research Turbine (CART), a research wind-turbine in use at the National Renewable Energy Laboratory (NREL [7]) for experimental purposes.

The box containing a picture of the aerogenerator on the left represents the multibody model of the wind-turbine, which accounts for its kinematics, structural dynamics and aerodynamics. The multibody model outputs the angular speed of the shaft, which is fed to the control system. The control modeling environment takes care of:

- generating the gusty airstream input for the multibody analysis
- determining the generator's torque to be applied to the shaft of the wind-turbine
- computing the blade pitch input to be used by the multibody model in order to control the behavior of the wind-turbine

The latter item represents the control task. Given the relative simplicity of the controller considered in this work, the whole problem could have been modeled monolithically within the general-purpose multibody simulation environment. Nonetheless, the control-related part has been intentionally modeled in a separate general-purpose, graphically driven mathematical modeling environment, for the sake of generality. This kind of graphical environment represents the natural modeling environment for control systems, and is offered by many popular software packages, e.g. The Mathworks' Matlab/Simulink [8], INRIA's Scilab/Scicos [9, 10], Labview [11], and National Instrument's MATRIXx [12].

The multibody analysis, instead, represents an effective means to provide the virtual simulation of the real process that needs to be controlled. For this reason, it only contains the bare physical process, in order to allow to test the real controller that would be used in a real-world application. The two processes typically communicate by means of real-time capable network primitives, or by inter-process communication when running on the same computer, on separate CPUs in case of SMP architectures.

There exists a number of software that can be proficiently used to analyze the aeroservoelastic behavior of wind-turbines. Some are dedicated to this task, while others are general-purpose. A recent survey of some of them is presented in [13]. That reference compares software based on accuracy with respect to benchmark problems. MBDyn has been coupled to NREL's AeroDyn library in order to exploit the availability of a well-proven wind-turbine aerodynamic code [14]. However, there are other factors that may come into play, significantly those related to software accessibility, to the capability of modeling problems with an arbitrary level of detail, and to fulfill control design requirements.

The problem of accessibility is addressed by using 'free software'. Problems can be analyzed with an arbitrary level of detail when general-purpose software is used. Control design requirements, and significantly the capability to perform hardware-in-the-loop virtual testing, are met by enabling tight real-time scheduling and execution of the simulation and control software. The proposed virtual testing environment meets all the requirements illustrated above.

The use of general-purpose multibody software typically results in solving larger problems, especially when the formulation is based on the redundant coordinate approach. This may represent a challenge for real-time simulation; for this reason, real-time simulations are often approached reducing the problem to a minimal set of coordinates. The redundant coordinate approach, however, usually results in very sparse problems. When sparsity is efficiently handled by specialized linear algebra solvers, as the one proposed in [15], very good performances can be obtained in terms of computational time. A detailed comparison of the effects of different linear solvers on the overall efficiency of real-time multibody simulation is presented in [16]. Complex mechanical systems, including robots and rotorcraft wind-tunnel models, can be simulated in real-time with the desired accuracy using general-purpose multibody software, with an acceptable trade-off between model detail and real-time implementation [17–19]. This is the case of MBDyn [20, 21], the free multibody software used and adapted for the purpose of this research.

### 3 Wind-Turbine Description

The CART wind-turbine, shown in Fig. 2, is located at the National Wind Technology Center (NWTC) of NREL in Colorado. It is used as state of the art test-bench for controls research in wind-engineering [22–25]. The main focus is currently on testing control strategies to improve the performances and the handling of wind-turbines subjected to exceptional operational conditions. It is an upwind machine with a nacelle tilt of 3.8 deg and two teetering blades with zero precone.



**Fig. 2** The Controls Advanced Research Turbine (CART)



The rotor diameter and the hub height are respectively 43.3 and 36.6 m. Power energy is rated at 600 kW and generator speed through a gearbox with a ratio of 43.165 is rated at 1800 rpm. The rotor is thus rated at an angular speed  $\Omega_{\text{rated}} = 41.7$  rpm.

This model has been selected because it is described in detail in the above mentioned publicly accessible documents. Moreover, since it is characterized by a two-blade, teeter rotor, it requires less computational effort than more modern, three-blade turbines. As illustrated in the following, this choice was conservative, as the proposed analysis leaves margin for further increase in model complexity without violating the real-time requirements.

## 4 Baseline Controller

The baseline controller is composed by independent electric torque and collective pitch algorithms. Both controllers use the rotor angular speed measurement as sole input.

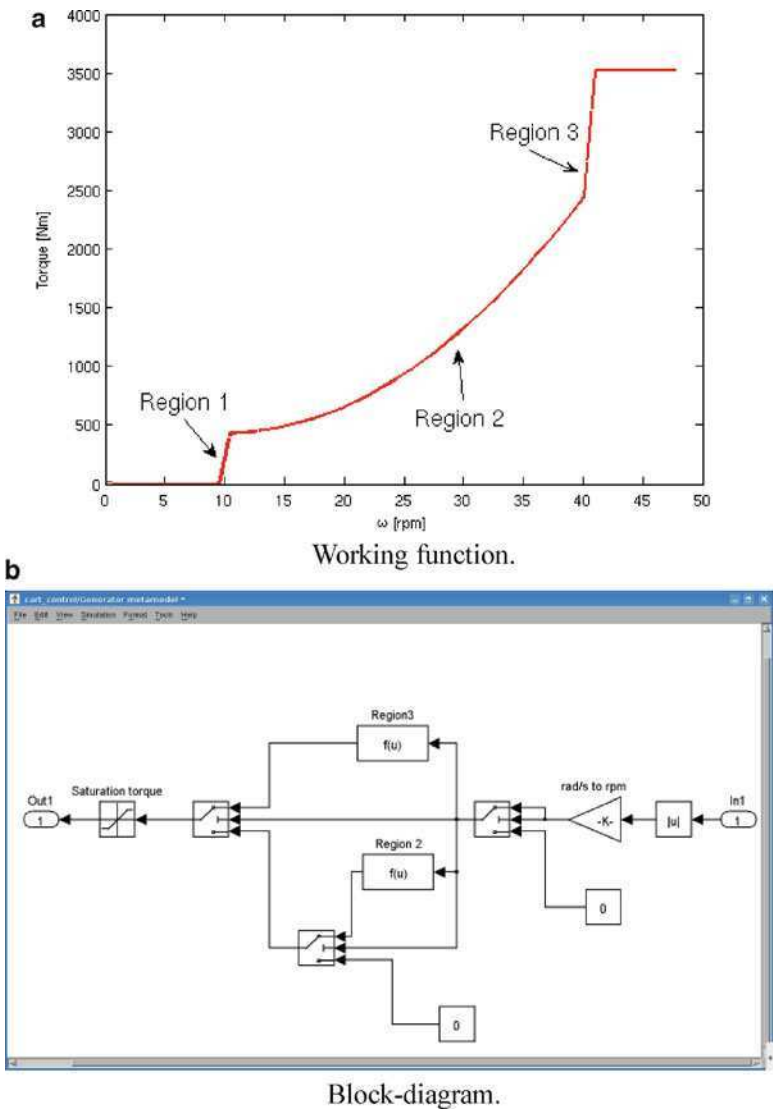
The task of the control-system is to maximize power capture below, and regulate a constant speed above, the rated operating point. Currently, no effort is undertaken to regulate the high speed generator shaft brake nor the nacelle yaw (which, in the real wind turbine, is limited to only  $\pm 0.5$  deg and simply used for tracking relatively small wind changes).

Generator commands are calculated by means of a piece-wise function. Below the cut-in speed of 10 rpm (Region 1), no electric torque is generated, to let the wind accelerate the rotor at maximum angular acceleration. The quadratic region (Region 2) is designed to keep the tip-speed ratio at the optimal value for maximum power. Above 99% of the rated rotor speed  $\Omega_{\text{rated}}$  a constant torque of 3524 Nm is required. Between 98% and 99% of  $\Omega_{\text{rated}}$  the transition is linear, equivalent to a slip of 5% (Region 3) [26–28]. Figure 3a shows the piece-wise working function for the electric generator (top), and its block-diagram model (bottom).

The full-span collective blade pitch angle commands are computed by means of a PID controller on the error of the rated angular speed, with saturation on the integral term to limit wind-up. Special care is taken to avoid working in post-stall regions during the initial acceleration phase. This allows to use the same controller for both rotor start-up and speed control at the rated speed.

Figure 4 shows the controlled system in the typical Scicos/Simulink/MATRIXx environment. The box labeled ‘from MBDyn’ represents the output of the multibody model that is input in the control system: the basic control system considered in this work requires only a measure of the angular velocity of the rotor. The box labeled ‘to MBDyn’ represents the inputs to the multibody model: the free-stream wind velocity, the torque absorbed by the electric generator and the desired blade pitch.

The controlled CART model can be run either in batch or real-time mode. In the latter case, the real-time scheduling can be delegated to the operating system using a standard POSIX interface, or tightly enforced using RTAI. When executing



**Fig. 3** Electric generator piecewise working function and block-diagram model

in batch mode, a generic interface between MBDyn and the external control software, based on standard UNIX inter-process communication primitives (local and TCP/IP sockets), is used. When executing in hard real-time mode, the controller code is automatically generated by any of Scicos, Simulink or MATRIXx from the very same model, and run by the RTAI operating system. In this case, MBDyn is scheduled in real-time by RTAI as well, to emulate the real wind-turbine.

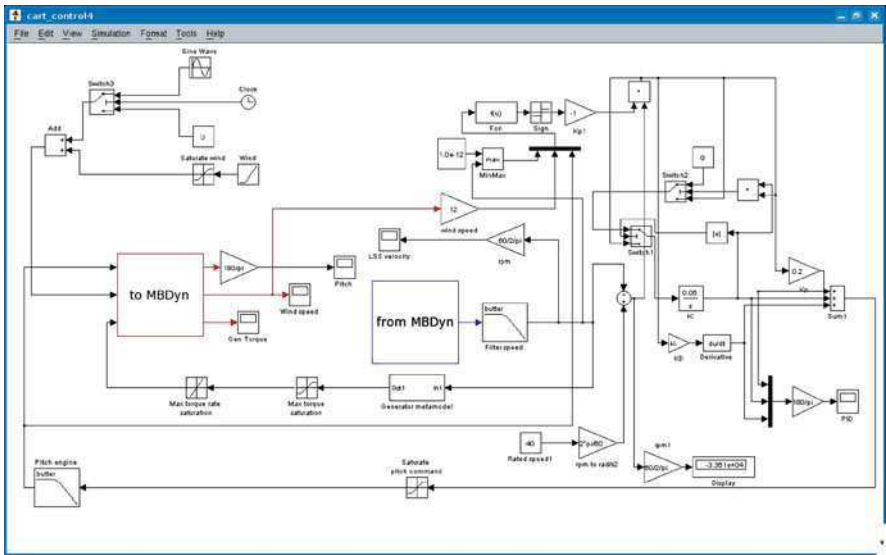


Fig. 4 Controlled system in the typical Scicos/Simulink/MATRIXx environment

In real time mode, the RTAILab graphical user interface [29] can be used to monitor the controller, either locally or remotely, and to tune gains and other system parameters on the fly, as shown later in an example.

5 Multibody Model

The multibody approach is definitely suited for the analysis of complex multidisciplinary systems where exact mechanism kinematics, nonlinear structural dynamics, and arbitrary control-related components need to be simultaneously analyzed [30, 31]. Thanks to its versatility and availability, the multibody formalism proposed in this work is being used in many fields related to aeroservoelasticity.

The analysis is based on an original formulation, implemented in the free general-purpose software MBDyn [4]. It performs the direct time integration of Initial Value Problems (IVP) written as a system of first-order Differential–Algebraic Equations (DAE), using implicit A/L stable integration algorithms [32].

5.1 Unconstrained Dynamics

The equations of motion of each unconstrained body are written in first order form using the Newton–Euler approach. The definitions of momentum,  $\beta_i$ , and momenta

moment about the node,  $\gamma_{\mathbf{x}_i}$ , for the  $i$ th node are

$$m_i \dot{\mathbf{x}}_i + \omega_i \times \mathbf{s}_{i\mathbf{x}_i} = \beta \quad (1a)$$

$$\mathbf{s}_{i\mathbf{x}_i} \times \dot{\mathbf{x}}_i + \mathbf{J}_{i\mathbf{x}_i} \omega_i = \gamma_{i\mathbf{x}_i} \quad (1b)$$

where  $m_i$  is the mass of the body connected to the  $i$ th node,  $\mathbf{x}_i$  is the location of the node,  $\omega_i$  is the angular velocity of the node,  $\mathbf{s}_{i\mathbf{x}_i}$  is the static moment of the body referred to the node's location, and  $\mathbf{J}_{i\mathbf{x}_i}$  is the inertia tensor of the body referred to the node's location. The equilibrium of each node yields

$$\dot{\beta}_i = \sum \mathbf{f}_i \quad (2a)$$

$$\dot{\gamma}_{i\mathbf{x}_i} + \dot{\mathbf{x}}_i \times \beta_i = \sum \mathbf{m}_{i\mathbf{x}_i} \quad (2b)$$

where all external forces,  $\mathbf{f}_i$ , and moments,  $\mathbf{m}_{i\mathbf{x}_i}$ , acting on the node are considered. The external forces and moments can arbitrarily depend on the motion of all nodes the  $i$ th one is connected to. The equations of motion of all the unconstrained nodes can be summarized as

$$\mathbf{M}\dot{\mathbf{q}} = \mathbf{p} \quad (3a)$$

$$\dot{\mathbf{p}} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{p}, t) \quad (3b)$$

where  $\mathbf{q} \in \mathbb{R}^n$  summarizes the kinematic variables of the nodes ( $n$  corresponds to 6 times the number of bodies,  $n_b$ ), while  $\mathbf{p} \in \mathbb{R}^n$  summarizes the momentum and momenta moments. The function  $\mathbf{f} : \mathbb{R}^{3n+1} \mapsto \mathbb{R}^n$  represents the generic configuration-dependent forces acting on the nodes. It includes the contributions related to structural deformability.

## 5.2 Constrained Dynamics

The constrained system dynamics are modeled by explicitly adding kinematic constraints between the nodes in form of algebraic equations, using Lagrange's multipliers formalism. The addition of  $m_h$  holonomic and  $m_{nh}$  non-holonomic constraints, respectively expressed by  $\phi(\mathbf{q}, t) = \mathbf{0} : \mathbb{R}^{n+1} \mapsto \mathbb{R}^{m_h}$  and  $\psi(\dot{\mathbf{q}}, \mathbf{q}, t) = \mathbf{0} : \mathbb{R}^{2n+1} \mapsto \mathbb{R}^{m_{nh}}$ , results in

$$\mathbf{M}\dot{\mathbf{q}} = \mathbf{p} \quad (4a)$$

$$\dot{\mathbf{p}} + \phi_{/\mathbf{q}}^T \lambda + \psi_{/\dot{\mathbf{q}}}^T \mu = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{p}, t) \quad (4b)$$

$$\phi(\mathbf{q}, t) = \mathbf{0} \quad (4c)$$

$$\psi(\dot{\mathbf{q}}, \mathbf{q}, t) = \mathbf{0} \quad (4d)$$

where  $\lambda \in \mathbb{R}^{m_h}$  and  $\mu \in \mathbb{R}^{m_{nh}}$  respectively are the multipliers related to the holonomic and non-holonomic constraints.

### 5.3 Structural Flexibility

The modeling of structural flexibility is fundamental for accurate aeroelastic analysis. However, accuracy may require a considerable number of degrees of freedom. As a consequence, accuracy may need to be traded for efficiency, especially when real-time simulation is considered.

Conventional horizontal axis wind turbines are characterized by a slender tower and very slender blades. The slenderness of current large size turbines will probably increase further as the size grows from 2 ÷ 5 MW on.

Historically, structural flexibility has been considered in multibody dynamics using lumped components first. Eventually, the need to bring the level of detail of finite elements led to combining the arbitrary reference rigid body motion peculiar of multibody dynamics with small perturbed deformation given by linear finite elements into the so-called floating frame approach [33]. In this case, a Ritz-like linear combination of deformation shapes is used to express a deformation with respect to a reference frame that undergoes arbitrary motion. This approach, in the case of wind turbines, suffers from the fact that an accurate basis consisting in normal vibration modes may require a significant number of degrees of freedom, since the normal modes are considerably influenced by the centrifugal stiffening [34]. This may not be an issue for systems rotating at constant angular velocity, but in general wind turbines can operate at an arbitrary velocity, and transient analysis capability is essential.

Accurate modeling of structural components can be achieved using finite elements directly in the multibody model [35]. The behavior of slender structural components can be efficiently described by the beam model. In many cases the beam model is fairly accurate and at the same time synthetic; thus, it leads to efficient models, allowing to meet real-time simulation requirements. In this work, an original, geometrically exact, composite-oriented beam formulation based on a finite-volume approach is used [36]. The beam model in the multibody analysis takes care of the one-dimensional flexibility of slender structural components. In order to give accurate results, it requires a correct and accurate characterization of the cross-section inertial and structural properties. In the present work, this pre-processing step is based on the section characterization procedure first proposed by Giavotto et al. [37], that allows to characterize the  $6 \times 6$  stiffness matrix of a generalized Timoshenko beam. A detailed review of different beam section characterization procedures is presented in [38]. Other formulations, including the one proposed in [39], are compared to standard wind turbine blade characterization approaches in [40].

### 5.4 Numerical Integration

The implicit DAE problem of (4) can be written in the generic form

$$\mathbf{g}(\dot{\mathbf{y}}, \mathbf{y}, t) = \mathbf{0}, \quad (5)$$

where  $\mathbf{y} = \{\mathbf{q}; \mathbf{p}; \lambda; \mu\}$  summarizes all the variables of (4). Its solution at the generic time step  $t_k$ , using a generic implicit multistep integration scheme, requires to solve (5) for  $\dot{\mathbf{y}}_k$ , with

$$\mathbf{y}_k = \sum_{r=1,n} a_r \mathbf{y}_{k-r} + h \sum_{s=0,n} b_s \dot{\mathbf{y}}_{k-s}. \quad (6)$$

The coefficients  $a_r$  and  $b_s$  characterize the numerical integration method;  $b_0 \neq 0$  for implicit schemes. Equation (5) is solved using a Newton–Raphson scheme, namely

$$\mathbf{g}_{/\dot{\mathbf{y}}} \delta \dot{\mathbf{y}}_k + \mathbf{g}_{/\mathbf{y}} \delta \mathbf{y}_k = -\mathbf{g}. \quad (7)$$

According to (6),  $\delta \mathbf{y}_k = h b_0 \delta \dot{\mathbf{y}}_k$ ; as a consequence, (7) yields

$$(\mathbf{g}_{/\dot{\mathbf{y}}} + h b_0 \mathbf{g}_{/\mathbf{y}}) \delta \dot{\mathbf{y}}_k = -\mathbf{g}. \quad (8)$$

The linear problem of (8) needs to be solved iteratively to convergence. The DAE nature of the problem implies that either of matrices  $\mathbf{g}_{/\dot{\mathbf{y}}}$ ,  $\mathbf{g}_{/\mathbf{y}}$  can be structurally singular, or both. However, when the problem is well posed, the matrix pencil  $(\mathbf{g}_{/\dot{\mathbf{y}}} + \lambda \mathbf{g}_{/\mathbf{y}})$  is not structurally singular; thus, (8) can be solved [41].

MBDyn is mainly used by its developers to model the aeroservoelasticity of rotary wing aircraft (e.g. [42]). Analysis of wind-turbine systems is carried out by some of the independent users [14,43].

## 5.5 CART Wind Turbine Multibody Model

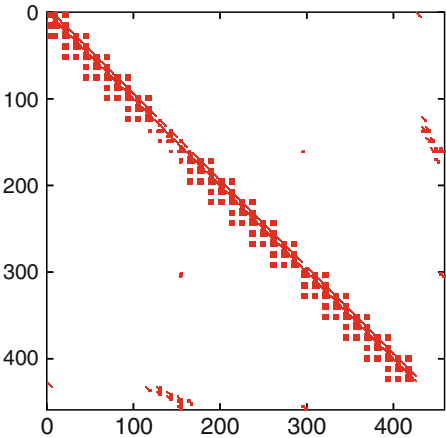
The multibody model considered in this work consists in:

- a deformable tower, made of five three-node finite-volume beam elements, clamped to the ground at the lower extremity
- a rigid nacelle, connected to the tower by a yaw hinge, with built-in pitch; the yaw degree of freedom in the analysis is restrained by a very stiff spring, since no yaw control is considered
- a rigid low-speed shaft, connected to the nacelle by ideal bearings; the rotational inertia includes that of the high-speed shaft, accounting for the low- to high-speed shaft gear ratio
- an ideal generator, consisting in an internal torque applied between the nacelle and the low-speed shaft, whose value is computed by the control task
- a rigid body that models the teetering hub, connected to the low-speed shaft by an ideal teeter joint
- a pair of deformable blades, modeled with 5 three-node finite-volume beam elements each, including blade element aerodynamics coupled to an induced flow model; each blade is hinged to the teetering body by means of a revolute hinge

**Table 1** Summary of CART model

Component	Nodes	Joints	Bodies	Beams	Aero	Forces	DoFs
Tower	11	1	10	5			138
Nacelle	1	2	1				17
Shaft	1	1	1				17
Generator						1	
Teeter	1	1	1				17
Blades, 2×	11	1	11	5	5		138
Total	36	7	35	15	10	1	465

**Fig. 5** Non-zero coefficients of the CART model matrix; fill-in is about 3%



that allows to impose the pitch angle. Although the pitch of each blade can be controlled independently, in this study the same value is applied for simplicity.

Table 1 summarizes the number of nodes, elements and degrees of freedom of each component of the model. It results in a total of 465 equations: 432 related to the dynamics of the nodes, and the remaining 33 related to holonomic constraints. The typical sparsity pattern of the matrix pencil of (8) is shown in Fig. 5. The number of non-zero coefficients is 6351, which implies a fill-in less than 3%. The matrix is intrinsically non-symmetric.

The blade pitch is controlled by simultaneously rotating the blades at the root node by an amount that is determined by the controller task. An angular velocity sensor measures the low-speed shaft velocity and feeds it into the control task.

The wind-turbine pictures in Fig. 6 are generated by an enhanced version of the free visualization software EasyAnim [44] (the modified version is available in [4]).

The structural model of the tower and of the blades is likely too refined, considering the very low rotational velocity and the bandwidth of interest. Figure 7a illustrates the convergence on the frequency of the first 10 modes of the tower plus nacelle model as the discretization is refined from 1 to 10 three-node beam elements. Figure 7b refers to the entire model at null angular velocity, with the blades in the vertical position. Similar trends are shown by all subcomponents.

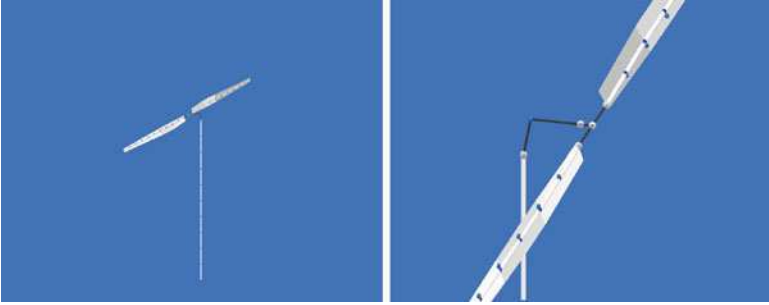


Fig. 6 Graphical representation of the CART model

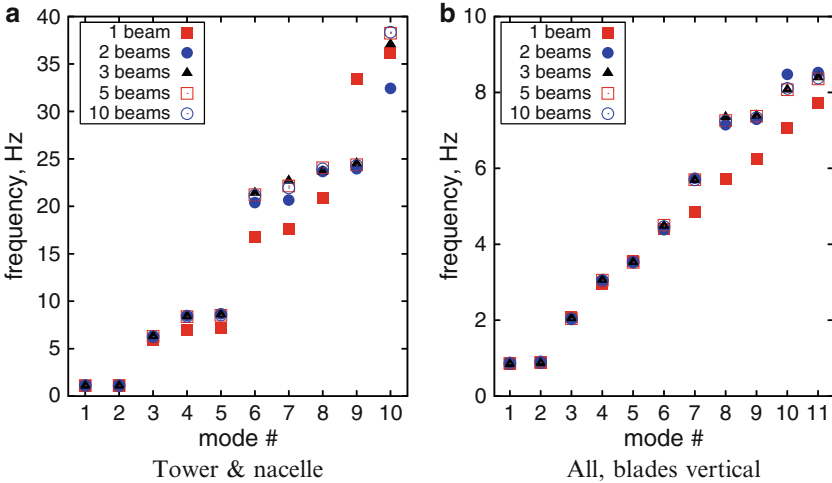


Fig. 7 CART model convergence on the first normal modes (non-rotating)

Five beam elements for the tower and five for each blade have been used because, as shown in Fig. 7, they yield a dynamically well converged model, while allowing to meet the real-time execution constraint.

## 6 Real-Time Simulation

The proposed multibody analysis runs in real-time thanks to RTAI support, built-in in MBDyn when running on Linux [17, 45, 46].

Popular graphical tools for computer-assisted control system design and fast prototyping, with automatic control code generation, like Scicos, Simulink, and MATRIXx, have been extended to support the generation of the controller source code in the C programming language, using RTAI's primitives for real-time scheduling and inter-process communication.



As a result, two real-time processes, one simulating the physical system and the other performing the appropriate control task, are executed on computers running the RTAI real-time extension for Linux.

The RTOS takes care of scheduling both processes with the desired periodicity. The processes typically communicate a set of measurements from MBDyn to the controller, and a set of control inputs from the controller to MBDyn. Inter-process communication uses RTAI mailboxes, a primitive that transparently uses shared memory when both tasks are executed on the same machine, or the UDP real-time support, provided by the NetRPC extension, when tasks are distributed on different machines.

In more sophisticated applications, a real-time instance of MBDyn itself can be embedded in the controller. In those cases, it is used to determine the control inputs required by the controlled process, which in turn can be a real or a simulated process [47].

Two scheduling approaches can be followed. In one case, the processes synchronize with each other by using RTAI semaphores. One process, usually the controller, is scheduled periodically. As soon as it sends the control input to the simulator, the simulator is woken up and starts simulating the time step. This approach guarantees that the subsequent time step receives the expected control input. In the other case, the two processes are independently scheduled periodically. Each process reads inputs and writes outputs according to its schedule. There is no strict guarantee that each process receives exactly the expected input. However, the error can be at most one sample period, and thus is treated as a disturbance.

The simulation must behave in a quasi-deterministic manner or, in other words, each sample interval needs to be completed within a given number of operations. This is not guaranteed when iteratively solving a nonlinear problem. In order to obtain a quasi-deterministic behavior, RT-MBDyn solves the nonlinear problem within up to a fixed number of iterations, using a modified Newton–Raphson scheme that consists in assembling and factorizing the matrix only at the first iteration of each time step. Errors due to lack of convergence to the desired accuracy can be reasonably assumed to be small after few iterations, thanks to the superlinear convergence properties of the modified Newton–Raphson scheme, provided the prediction at each time step is close enough to the actual solution. These errors can be treated as disturbances by the control scheme. Sporadic overruns can be accepted as disturbances, provided subsequent steps can “catch up” with the controller.

Figure 8 shows a fairly broad layout of the real-time simulation setup, where the simulation and the controller are located on different computers connected by a hard real-time network via NetRPC, while multiple supervising stations monitor the output of the controller and of the simulation using soft real-time connections, with the possibility of optionally modifying the controller’s parameters.

Figure 9 shows the result in terms of rotor angular speed and pitch command of a simulation in correspondence of a growing wind speed rated at an average level of 12 m/s. In detail, Fig. 9a refers to a random disturbance of 20% of the wind velocity magnitude, while Fig. 9b refers to a sinusoidal disturbance whose amplitude is 5% the wind velocity magnitude. In both cases, the resulting error on the final rated

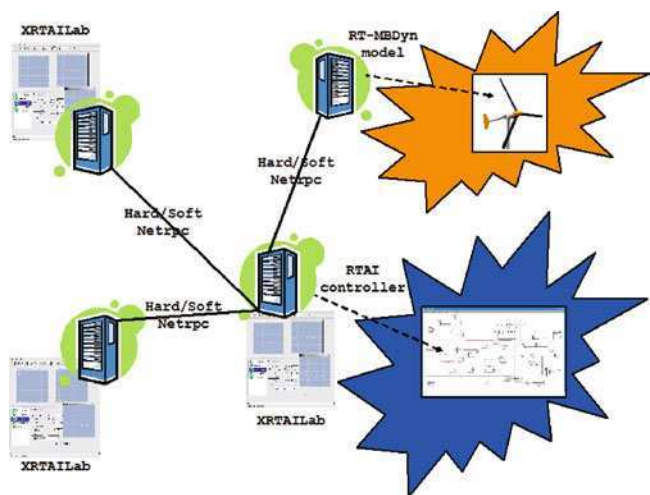


Fig. 8 Sketch of a generic distributed real-time simulation layout

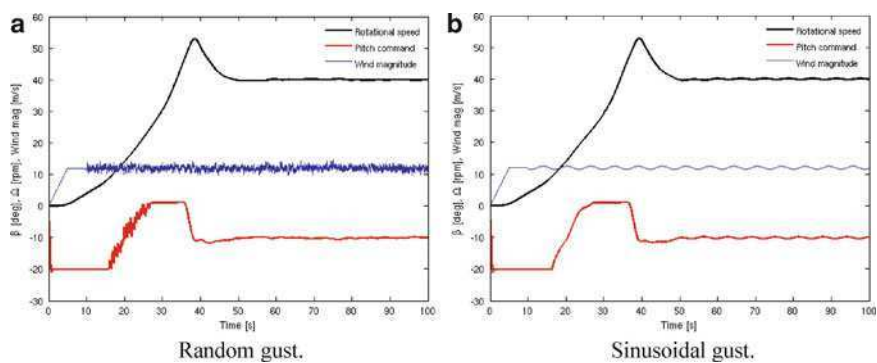
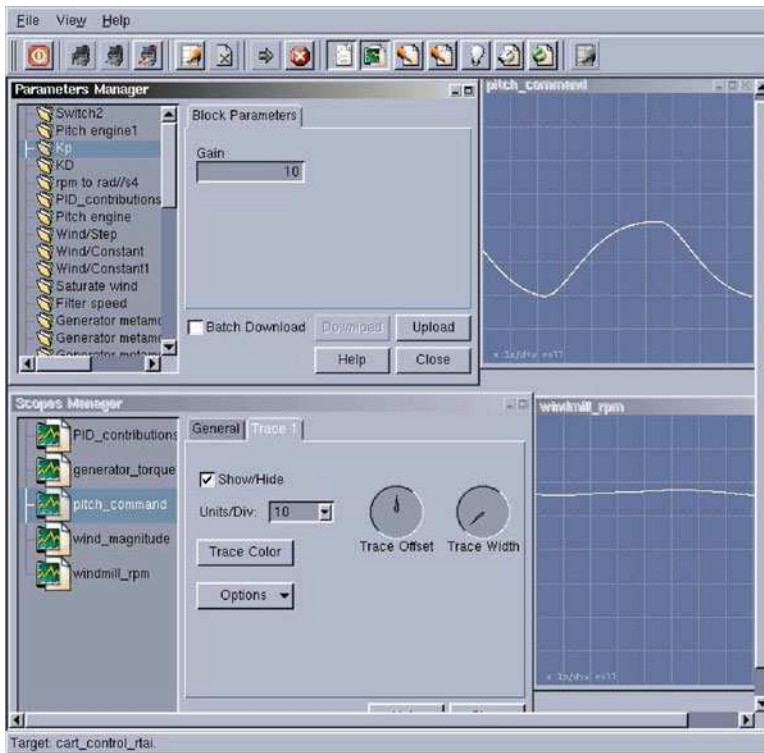


Fig. 9 Rotor angular speed and pitch command for wind-up and gusty wind

angular speed is less than 0.5%. The sample rate is 100Hz. Figure 10 shows the output of the controlled CART model within the RTAILab environment. The field labeled as ‘Gain’ in the top left portion of the control panel allows to change the controller’s parameters while the simulation is running in real-time. The control panel can be configured to allow access to any of the control parameters that are exported when the control system is designed.

The numerical simulations have been performed on a Dual Core AMD Opteron Processor 280 (1 GHz). In all cases the multibody model could be executed well within the required sample rate of 100Hz. This leaves room for further model refinement, e.g. an increase of the sampling rate, or the analysis of more complex turbines, e.g. three-bladed.



**Fig. 10** Output of the controlled CART model within the RTAILab environment

One of the distinguishing features of the proposed approach is that information related to distributed structural flexibility can be simulated and monitored in real-time. This paves the way to simulating in real-time the control of strains, stresses, and gust load reduction in general.

## 7 Conclusions

This work illustrates the implementation of what can essentially be considered a test bench to prove the feasibility of innovative, efficient and low-cost solutions for fast-prototyping and customization of controlled mechanical and aeroservoelastic systems. The proposed environment has been applied to the development and testing of a simple controller for wind-turbines. Further details can be added, both in the simulated physical process, to enhance system modeling with features that have not been considered so far in this work, and in the controller, to investigate more sophisticated control strategies.

## 8 Additional Material

The analysis was performed using RTAI 3.6.1, available for download at [3], and MBDyn 1.3.3, available for download at [4]. The wind-turbine models and the controller source code are available at [4], in the RT-MBDyn→wind turbine folder. Feedback using the mailing lists *rtai@rtai.org* and *mbdyn-users@mbdyn.org* is appreciated.

**Acknowledgements** This work has been partially funded by ‘SI PARTE!’, a R&D project of ‘Regione Lombardia’ that addressed the development of innovative solutions for Embedded Real-Time Applications.

## References

1. Hansen MOL, Sørensen JN, Voutsinas S, Sørensen N, Madsen HAa (2006) State of the art in wind turbine aerodynamics and aeroelasticity. *Progr Aero Sci* 42(4):285–330
2. Barlas TK, van Kuik GAM (2009) Review of state of the art in smart rotor control research for wind turbines. *Progr Aero Sci* 46(1):1–27
3. Real-time application interface (RTAI). <http://www.rtai.org/> (last accessed March 2009)
4. Multibody dynamics (MBDyn). <http://www.aero.polimi.it/mbdyn/> (last accessed March 2009)
5. Garsch R, Twele J (2002) Wind power plants, fundamentals, design, construction, and operation. In: James and James (ed) Solarpraxis, Berlin
6. Jonkman J, Butterfield S, Musial W, Scott G (2009) Definition of a 5-MW reference wind turbine for offshore system development. NREL/TP-500-38060
7. National renewable energy laboratory (NREL). <http://www.nrel.gov/> (last accessed November 2009)
8. Matlab. <http://www.mathworks.com/> (last accessed November 2009)
9. Scilab. <http://www.scilab.org/> (last accessed November 2009)
10. Scicos. <http://www.scicos.org/> (last accessed November 2009)
11. NI LabVIEW. <http://www.ni.com/labview/> (last accessed November 2009)
12. NI MATRIXx. <http://www.ni.com/matrixx/> (last accessed November 2009)
13. Buhl ML Jr, Manjock A (2006) A comparison of wind turbine aeroelastic codes used for certification. In: 44th AIAA aerospace sciences meeting and exhibit, Reno, Nevada, USA, 9–12 January
14. Meng F, Masarati P, van Tooren M (2009) Free/open source multibody and aerodynamic software for aeroservoelastic analysis of wind-turbines. In: 47th AIAA aerospace sciences meeting, Orlando, Florida, USA, 5–8 January
15. Morandini M, Mantegazza P (2007) Using dense storage to solve small sparse linear systems. *ACM Trans Math Software* 33(1), doi:10.1145/1206040.1206045
16. González M, González F, Dopico D, Luaces A (2008) On the effect of linear algebra implementations in real-time multibody system dynamics. *Comput Mech* 41(4):607–615
17. Attolico M, Masarati P (2003) A multibody user-space hard real-time environment for the simulation of space robots. In: 5th real-time linux workshop, Valencia, Spain, 9–11 November
18. Masarati P, Attolico M, Nixon MW, Mantegazza P (2004) Real-time multibody analysis of wind-tunnel rotorcraft models for virtual experiment purposes. In: AHS 4th decennial specialists’ conference on aeromechanics, Fisherman’s Wharf, San Francisco, CA, 21–23 January
19. Attolico M, Masarati P, Mantegazza P (2005) Trajectory optimization and real-time simulation for robotics applications. In: Multibody dynamics 2005, ECCOMAS thematic conference, Madrid, Spain, 21–24 June

20. Masarati P, Morandini M, Quaranta G, Mantegazza P (2003) Open-source multibody analysis software. In: Multibody dynamics 2003, international conference on advances in computational multibody dynamics, Lisboa, Portugal, 1–4 July
21. Masarati P, Morandini M, Quaranta G, Mantegazza P (2005) Computational aspects and recent improvements in the open-source multibody analysis software “MBDyn”. In: Multibody dynamics 2005, ECCOMAS thematic conference, Madrid, Spain, 21–24 June
22. Fingersh LJ, Johnson K (2002) Controls advanced research turbine (CART) commissioning and baseline data collection. NREL/TP-500-32879
23. Stol KA (2003) Geometry and structural properties for the controls advanced research turbine (CART) from model tuning. NREL/SR-500-32087
24. Wright AD, Fingersh LJ, Balas MJ (2006) Testing state-space controls for the controls advanced research turbine. In: 44th AIAA aerospace sciences meeting and exhibit, Reno, Nevada, USA, 9–12 January
25. Wright AD, Fingersh LJ, Stol KA (2007) Designing and testing controls to mitigate tower dynamic loads in the controls advanced research turbine. In: 45th AIAA aerospace sciences meeting and exhibit, wind energy symposium, Reno, Nevada, USA, 8–11 January
26. Stol KA, Fingersh LJ (2003) Wind turbine field testing of state-space control designs. NREL/SR-500-35061
27. Fuglseth TP (2005) Modelling a 2.5 MW direct driven wind turbine with permanent magnet generator. Technical report, Department of Electrical Power Engineering. Norwegian University of Science and Technology. <http://www.elkraft.ntnu.no/smola2005/Topics/9.pdf>
28. Hansen MH, Hansen A, Larsen TB, Øye S, Sørensen P, Fuglsang P (2005) Control design for a pitch regulated, variable speed wind turbine. RISØ R-1500(EN), Denmark
29. Bucher R, Dozio L (2003) CACSD under RTAI linux with RTAI-LAB. In: 5th real-time linux workshop, Valencia, Spain, 9–11 November
30. Lee D, Hodges DH (2004) Multi-flexible-body analysis for application to wind turbine control design. NREL/SR-500-35228
31. Bottasso CL, Croce A, Savini B, Sirchi W, Trainelli L (2006) Aero-servo-elastic modeling and control of wind turbines using finite-element multibody procedures. Multibody Syst Dyn 16:291–308
32. Masarati P, Lanz M, Mantegazza P (2001) Multistep integration of ordinary, stiff and differential-algebraic problems for multibody dynamics applications. In: XVI Congresso Nazionale AIDAA, Palermo, 24–28 September, pp 71.1–10
33. Shabana AA (1997) Flexible multibody dynamics: Review of past and recent developments. Multibody Syst Dyn 1(2):189–222
34. Wallrapp O, Schwertassek R (1991) Representation of geometric stiffening in multibody system simulation. Int J Numer Meth Engng 32:1833–1850
35. Géradin M, Cardona A (2001) Flexible multibody dynamics: A finite element approach. Wiley, Chichester
36. Ghiringhelli GL, Masarati P, Mantegazza P (2000) A multi-body implementation of finite volume beams. AIAA J 38(1):131–138
37. Giavotto V, Borri M, Mantegazza P, Ghiringhelli GL, Caramaschi V, Maffioli GC, Mussi F (1983) Anisotropic beam theory and applications. Comput Struct 16(1–4):403–413
38. Hodges DH (2006) Nonlinear Composite Beam Theory. AIAA - c2006 - XII. Reston, VA
39. Bauchau OA (1985) A beam theory for anisotropic materials. J Appl Mech 107:416–422
40. Chen H, Yu W, Capellaro M (2009) A critical assessment of computer tools for calculating composite wind turbine blade properties. Wind Energy, published online December 14, doi:10.1002/we.372
41. Brenan KE, Campbell SLA, Petzold LR (1989) Numerical solution of initial-value problems in differential–algebraic equations. North-Holland, New York
42. Quaranta G, Masarati P, Mantegazza P (2004) Assessing the local stability of periodic motions for large multibody nonlinear systems using POD. J Sound Vib, 271(3–5):1015–1038
43. Meng F, Pavel MD, van Tooren M (2008) Aeroelastic stability analysis of large scale horizontal axis wind turbines using reduced order system identification based on flexible nonlinear multibody dynamics. In: 46th AIAA aerospace sciences meeting and exhibit, Reno, Nevada, USA, 7–10 January

44. EasyAnim. <http://mecara.fpms.ac.be/EasyDyn/> (last accessed March 2009)
45. Morandini M, Masarati P, Mantegazza P (2005) A real-time hardware-in-the-loop simulator for robotics applications. In: Multibody dynamics 2005, ECCOMAS thematic conference, Madrid, Spain, 21–24 June
46. Morandini M, Masarati P, Mantegazza P (2005) Performance improvements in real-time general-purpose multibody virtual experimenting of rotorcraft systems. In: 31st European rotorcraft forum, Firenze, Italy, 13–15 September
47. Fumagalli A, Masarati P (2009) Real-time computed torque control using general-purpose multibody software. *Multibody Syst Dyn* 22(1):47–68

# Comparison of Planar Structural Elements for Multibody Systems with Large Deformations

Markus Dibold and Johannes Gerstmayr

**Abstract** In the field of multibody dynamics, structural components, such as beams or plates, have been discretized in different ways, according to special requirements of certain problem configurations. In literature, models, which follow the same mechanical theories but a different numerical discretization technique, such as the absolute nodal coordinate formulation (ANCF) and the floating frame of reference formulation (FFRF), have been calculated for comparison. In existing examples, the solutions of these calculations do not always coincide very accurately. Therefore, in the present contribution, which is an extension of a former work of the authors, standard static and dynamic problems in the large deformation regime are treated. Special emphasis is laid on converged solutions, using an analytical reference value in the static case. For dynamic examples a reference value based on the strain energy is provided, in order to simplify the comparison of the different formulations and to provide a reference value, similar to the static case, for future studies. For both formulations planar finite elements based on the Bernoulli–Euler theory are utilized. In case of the ANCF the finite element consists of two position and two slope coordinates in each node only. In the FFRF beam finite element, as usual, two sets of coordinates are used to describe the actual configuration. The first set of coordinates defines the location and orientation of the body reference frame. The second set of coordinates describes small superimposed transverse and axial deflections relative to the body frame. The transverse deflections are approximated by means of two static modes for the rotation at the boundary and a user-defined number of eigenmodes of the clamped-clamped beam. The axial deflection is represented by a linear approach. In numerical studies, the accuracy of the two formulations is compared for two example problems, a cantilever beam with a singular force at the free end and a slider-crank mechanism. It turns out that both formulations have comparable performance and that the results coincide in the converged case.

---

M. Dibold (✉) and J. Gerstmayr

Linz Center of Mechatronics GmbH (LCM), Altenbergerstraße 69, 4040 Linz, Austria

e-mail: [markus.dibold@lcm.at](mailto:markus.dibold@lcm.at); [johannes.gerstmayr@lcm.at](mailto:johannes.gerstmayr@lcm.at)

# 1 Introduction

For the numerical simulation of beam-type flexible multibody systems several methods have been established. Traditionally, these simulations are performed with a floating frame of reference formulation (FFRF). In this formulation the position or displacement of a point of a body is decoupled into a small (flexible) deformation, which is described in a local body-fixed coordinate system, and a possibly large superimposed rigid body motion. The relative displacements are described by means of a Ritz approximation, where the space-wise discretization is performed with eigenmodes or polynomials, see [1, 2]. The use of relative displacement kinematics consequently leads to a non-constant mass matrix and centrifugal and Coriolis terms, represented by the quadratic velocity vector. The virtual work of internal forces leads to a constant stiffness matrix in the case of small deformations. The numerical solution of these problems can be obtained using well-known numerical integration techniques, [3]. If geometric or material nonlinearities occur, the stiffness matrix becomes non-constant. The computational efficiency can be improved for systems with non-constant mass matrices using implicit time integration methods and simplified Jacobians [4].

For large deformation problems a pioneering paper was written by Simo and Vu-Quoc [5], where a fixed frame of reference was used, and the so-called geometrically exact formulation of the strain energy of the beam finite element was based on classical nonlinear rod theories. Neglecting shear deformation, this theory coincides with the strain definitions in the Euler elastica [6]. The numerical treatment of geometrically exact beam elements is based on discretization of the position of the beam centerline and rotation of the cross section. The strain energy can be interpreted with the help of a co-rotational formulation of the strain. Alternatively, in the absolute nodal coordinate formulation (ANCF), which is a large deformation finite element formulation, the deformation of the beam is interpolated on basis of position and slope degrees of freedom in the nodes. The use of slopes instead of rotational parameters still allows the exact representation of rotational inertia of a rigid body [7]. Originally, fully parameterized ANCF finite elements showed various types of locking, such as Poisson locking or combined shear and thickness locking. These effects could be resolved, and the solution of ANCF finite elements now leads to good agreement with the solution of fully three-dimensional computations with solid finite elements, see [8]. An important advantage of the absolute nodal coordinate formulation is the constant mass matrix. The elastic forces are nonlinear and usually derived from the deformation field taking into account the fully nonlinear Green strain tensor and the second Piola–Kirchhoff stress tensor [9]. As an alternative to fully parameterized ANCF elements, simplified ANCF elements have been developed, which only take into account the deformation of the elastic line of the beam element. This formulation is computationally more efficient, but less general. When restricting e.g. to the Bernoulli–Euler case, only nodal positions and axial slopes are utilized as degrees of freedom.

The present investigation is an extension of a former work of the authors [10], where nonlinear beam elements based on the fundamentally different approaches of



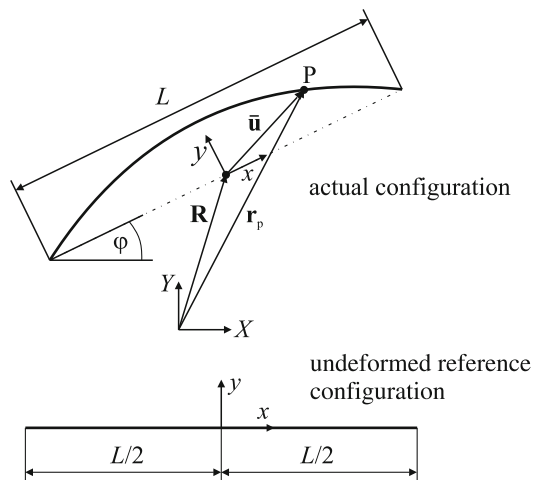
the FFRF and ANCF are compared for example problems in the large deformation regime. Both types of elements include axial deformation and bending. In case of the ANCF the deformations are intrinsic coupled, whereas in the FFRF they are uncoupled. To oppose the different formulations, a brief introduction to each of them is given. A comparison of plane static and dynamic example problems of the corresponding beam finite elements is performed and characteristic results are shown. The convergence of the results is studied with respect to significant values, in order to provide reference values for future studies. It is shown that the results of the FFRF and the ANCF coincide in the converged case, although the studied methods have a completely different numerical discretization.

## 2 Floating Frame of Reference Formulation

In the floating frame of reference formulation two sets of coordinates are used to describe the configuration of a flexible body [11]. The first set of coordinates defines the origin and the orientation of a reference frame with respect to the inertial frame. In the present work the reference frame is fixed to the flexible body such that both ends of the beam are located at the  $x$ -axis of the body frame. The body reference represents (possibly large) rigid body translation and rotation. The second set of coordinates describes small elastic deformations relative to the reference frame.

In the present contribution we restrict ourselves to the plane case, where each structure is represented by a specific number of beam finite elements. In the following, general assumptions are made for one finite element. The configuration of this element is shown in Fig. 1.

The origin of the reference  $(x, y)$  coordinate system is given by the position vector  $\mathbf{r}_0$  with respect to the inertial  $(X, Y)$  coordinate system. The orientation is



**Fig. 1** FFRF model of a finite element

determined by  $\varphi$ , the corresponding angle about the axis of rotation. It is noteworthy that the center of mass in the undeformed beam element is at the origin of the reference frame, which leads to simpler expressions in the kinetic energy. The following derivation is provided according to the widely known formulation as given in the monograph of Shabana [11]. Additionally, special emphasis is laid on the formulation of the specific element.

The elastic deformations of the finite element are modeled by means of the Bernoulli–Euler beam theory, including bending and axial deformation. The position vector of a point P at the deformed axis of the finite element is defined according to Fig. 1,

$$\mathbf{r}_p = \mathbf{r}_0 + \mathbf{A} \bar{\mathbf{u}} \quad (1)$$

where  $\mathbf{r}_0 = [r_{0X} \ r_{0Y}]^T$ . In Eq. (1)  $\bar{\mathbf{u}}$  represents the position of point P in the floating reference frame, given by

$$\bar{\mathbf{u}} = \bar{\mathbf{u}}_0 + \bar{\mathbf{u}}_f = \begin{bmatrix} x \\ 0 \end{bmatrix} + \begin{bmatrix} u(x, t) \\ w(x, t) \end{bmatrix}, \quad (2)$$

including the axial coordinate  $x \in [-L/2..L/2]$  and the axial and transverse deformation  $u(x, t)$  and  $w(x, t)$ . The position vector  $\bar{\mathbf{u}}$  is transformed into the global coordinate system by means of the rotation matrix  $\mathbf{A}$ , which only depends on the rigid body angle  $\varphi$ . The velocity vector is computed from Eq. (1)

$$\dot{\mathbf{r}}_p = \dot{\mathbf{r}}_0 + \dot{\mathbf{A}} \bar{\mathbf{u}} + \mathbf{A} \dot{\bar{\mathbf{u}}} \quad (3)$$

where  $(\dot{\phantom{x}}) = \partial/\partial t$  denotes the derivative with respect to time. The space-wise discretization is performed with a Ritz approach. This leads to the following approximations of the deflections

$$u(x, t) = \sum_{j=1}^N S_u^j(x) q_f^j(t) \text{ and } w(x, t) = \sum_{j=1}^M S_w^j(x) q_f^{j+N+1}(t) \quad (4)$$

with shape functions  $S_u^j(x)$  and  $S_w^j(x)$  and time-dependent elastic coordinates  $q_f^j(t)$ . In the following, a matrix notation for  $\bar{\mathbf{u}}_f$  will be derived, in order to get a readable formulation of Eq. (2), when inserting the approximations of the deformations. Therefore, the shape functions  $S_u^j(x)$  and  $S_w^j(x)$  are collected in two row vectors  $\mathbf{S}_1(x)$  and  $\mathbf{S}_2(x)$  of length  $(N + M)$ , such that

$$\begin{aligned} \mathbf{S}_1(x) &= \begin{bmatrix} S_u^1 & \dots & S_u^N & 0 & \dots & 0 \end{bmatrix} = \begin{bmatrix} S_1^1 & \dots & S_1^{N+M} \end{bmatrix} \\ \mathbf{S}_2(x) &= \begin{bmatrix} 0 & \dots & 0 & S_w^1 & \dots & S_w^M \end{bmatrix} = \begin{bmatrix} S_2^1 & \dots & S_2^{N+M} \end{bmatrix} \end{aligned} \quad (5)$$

In Eq. (5) it is assumed that the shape functions to describe axial and transversal deflections are not intrinsic coupled. The time dependant elastic coordinates  $q_f^j(t)$  consequently are collected in a column vector of the same length  $(N + M)$ . Inserting

Eq. (5) into Eq. (4) and the resulting formulations into Eq. (2) leads to a matrix notation for  $\bar{\mathbf{u}}_f$ , where

$$\bar{\mathbf{u}}_f = \begin{bmatrix} u(x, t) \\ w(x, t) \end{bmatrix} = \begin{bmatrix} \mathbf{S}_1(x) \\ \mathbf{S}_2(x) \end{bmatrix} \mathbf{q}_f(t) = \mathbf{S}(x) \mathbf{q}_f(t), \quad (6)$$

Since dynamics in axial direction include significantly higher frequencies than in transverse direction, the shape functions  $S_1^k, k \leq N$ , are derived from a linear approximation of the axial deformation,

$$S_1^1(x) = s_0^1 + s_1^1 x, \quad (7)$$

which leads to  $N = 1$ . This linear approximation of the axial deformation still gives satisfying results for the calculation of the axial deformation. Inserting the boundary conditions

$$S_1^1\left(x = -\frac{L}{2}\right) = 0 \text{ and } S_1^1\left(x = \frac{L}{2}\right) = 1 \quad (8)$$

into Eq. (8) leads to the corresponding shape function

$$S_1^1 = \frac{1}{2} + \frac{x}{L}. \quad (9)$$

The shape functions to describe the transversal deflection are approximated according to the reduction methods, see [12]. In this field, structural elements are described by means of the rotation at the boundary and a finite number of shape functions, which usually are found from the corresponding frequency equation. In the present case the static modes for the rotation at the boundary are approximated by a cubic polynomial, which leads to

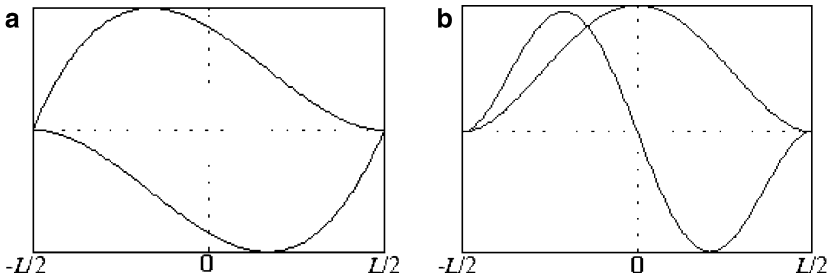
$$S_2^{N+1} = \frac{L}{8} - \frac{1}{4}x - \frac{1}{2L}x^2 + \frac{1}{L^2}x^3 \text{ and } S_2^{N+2} = -\frac{L}{8} - \frac{1}{4}x + \frac{1}{2L}x^2 + \frac{1}{L^2}x^3. \quad (10)$$

The shape functions are given by a user-defined number of eigenmodes,  $s$ , of the clamped-clamped beam. These eigenmodes,  $S_b^i, i = 1..s$ , follow from the partial differential equations of the beam finite element with

$$S_b^i(x) = (\cosh(\xi) - \cos(\xi)) - a (\sinh(\xi) - \sin(\xi)), \quad (11)$$

where the following parameters have to be taken into account

$$a = \frac{\cosh(L/\lambda^i) - \cos(L/\lambda^i)}{\sinh(L/\lambda^i) - \sin(L/\lambda^i)}, \quad \xi = \frac{(x + L/2)}{\lambda^i} \text{ and} \quad (12)$$



**Fig. 2** (a) Static modes and (b) first two eigenmodes of the FFRF finite element

$\lambda^i$ , which has to be solved from

$$\cosh(L/\lambda^i) \cos(L/\lambda^i) = 1. \quad (13)$$

The user-defined eigenmodes are finally included into the row vector of shape functions  $\mathbf{S}_2$  such that

$$\mathbf{S}_2^{N+2+i} = \mathbf{S}_b^i \quad (14)$$

and  $M = 2 + s$ . The static modes and the first two eigenmodes to approximate transversal deflections are shown in Fig. 2 respectively.

Due to the restriction to plane problems, the rotation matrix depends only on the rigid body angle  $\varphi$ . Therefore, the expression  $\dot{\mathbf{A}}\bar{\mathbf{u}}$  of Eq. (3) can be rewritten by

$$\dot{\mathbf{A}}\bar{\mathbf{u}} = \dot{\varphi} \mathbf{A}_\varphi \bar{\mathbf{u}} = \dot{\varphi} \mathbf{B}, \quad (15)$$

with the abbreviations  $\mathbf{A}_\varphi = \partial \mathbf{A} / \partial \varphi$  and

$$\mathbf{B} = \begin{bmatrix} -\sin \varphi & -\cos \varphi \\ \cos \varphi & -\sin \varphi \end{bmatrix} \begin{bmatrix} x + u(x, t) \\ w(x, t) \end{bmatrix}, \quad (16)$$

such that a matrix notation of Eq. (3) can be found by

$$\dot{\mathbf{r}}_P = [\mathbf{I} \quad \mathbf{B} \quad \mathbf{AS}] \begin{bmatrix} \dot{\mathbf{r}}_0 \\ \dot{\varphi} \\ \dot{\mathbf{q}}_f \end{bmatrix} = \mathbf{L} \dot{\mathbf{q}}. \quad (17)$$

In Eq. (17)  $\mathbf{I}$  describes a  $2 \times 2$  identity matrix,  $\mathbf{q}$  is the vector of generalized coordinates of one finite element and  $\mathbf{L} = [\mathbf{I} \quad \mathbf{B} \quad \mathbf{AS}]$ .

The equations of motion are usually derived from Lagrange's equations,

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\mathbf{q}}} \right)^T - \left( \frac{\partial T}{\partial \mathbf{q}} \right)^T + \left( \frac{\partial \mathbf{C}}{\partial \mathbf{q}} \right)^T \boldsymbol{\lambda} = \mathbf{Q} \quad (18)$$

where the kinetic energy  $T$ , internal and external forces  $\mathbf{Q} = \mathbf{Q}_i + \mathbf{Q}_e$ , as well as possible Lagrange multipliers  $\lambda$  due to constraint equations  $\mathbf{C} = 0$  need to be considered. The kinetic energy for the beam finite element is given by

$$T = \frac{1}{2} \rho A \int_{-L/2}^{L/2} \left( \dot{\mathbf{r}}_P^T \dot{\mathbf{r}}_P \right) dx \quad (19)$$

where  $\rho$  is the mass density and  $A$  is the cross-section of the beam element. By means of Eq. (17), the kinetic energy reads

$$T = \frac{1}{2} \dot{\mathbf{q}}^T \left( \rho A \int_{-L/2}^{L/2} \mathbf{L}^T \mathbf{L} dx \right) \dot{\mathbf{q}} = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} \quad (20)$$

including the mass matrix  $\mathbf{M}$  of a beam element. The single components of the mass matrix can be found by means of the shape functions and the generalized coordinates of the FFRF beam finite element, where all constant terms are pre-computed during the initialization of the finite element. For an extensive derivation of the single parts of the mass matrix see [13]. Evaluation of terms of Lagrange's equations, which contain kinetic energy leads to

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\mathbf{q}}} \right)^T - \left( \frac{\partial T}{\partial \mathbf{q}} \right)^T = \mathbf{M} \ddot{\mathbf{q}} + \dot{\mathbf{M}} \dot{\mathbf{q}} - \frac{1}{2} \left[ \frac{\partial}{\partial \mathbf{q}} \left( \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} \right) \right]^T = \mathbf{M} \ddot{\mathbf{q}} - \mathbf{Q}_v. \quad (21)$$

In Eq.(21) the so called quadratic velocity vector  $\mathbf{Q}_v$ , see Shabana et al. [7], is given as

$$\mathbf{Q}_v = -\dot{\mathbf{M}} \dot{\mathbf{q}} + \frac{1}{2} \left[ \frac{\partial}{\partial \mathbf{q}} \left( \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} \right) \right]^T. \quad (22)$$

The virtual work of internal forces in its general form reads,

$$\delta U = \int_V \boldsymbol{\sigma}^T \delta \boldsymbol{\varepsilon} dV \quad (23)$$

where  $\boldsymbol{\sigma}$ , is a vector gathering the six components of the stress tensor, and  $\boldsymbol{\varepsilon}$  is the corresponding vector of strain components. In case of a Bernoulli–Euler beam only the axial strain component  $\varepsilon_{xx}$  is taken into account. According to [14], the axial strain can be approximated by

$$\varepsilon_{xx} = u' - y w'' = (\mathbf{S}'_1 - y \mathbf{S}''_2) \mathbf{q}_f = \hat{\mathbf{S}} \mathbf{q}_f \quad (24)$$

in which  $(\cdot)' = \partial/\partial x$  and  $(\cdot)'' = \partial^2/\partial x^2$  denote differentiations with respect to the local coordinate  $x$  and the row vector  $\hat{\mathbf{S}} = (\mathbf{S}'_1 - y \mathbf{S}''_2)$ . Including Young's modulus  $E$ , the strain–stress relation is

$$\sigma_{xx} = E \varepsilon_{xx}. \quad (25)$$

Inserting Eqs. (24) and (25) into Eq. (23), the variation of the strain energy can be written

$$\delta U = \mathbf{q}_f^T \left( \int_V \hat{\mathbf{S}}^T \hat{\mathbf{S}} dV \right) \delta \mathbf{q}_f = \mathbf{q}_f^T \mathbf{K}_{ff} \delta \mathbf{q}_f, \quad (26)$$

in which the stiffness matrix  $\mathbf{K}_{ff}$  associated with the elastic coordinates of the finite element can be identified. Rewriting Eq. (26) by means of the element coordinates leads to

$$\delta U = \mathbf{q}^T \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \mathbf{K}_{ff} \end{bmatrix} \delta \mathbf{q} = \mathbf{q}^T \mathbf{K} \delta \mathbf{q} = \mathbf{Q}_i^T \delta \mathbf{q} \quad (27)$$

in which  $\mathbf{K}$  is the stiffness matrix of one beam element.

The virtual work of the external forces,  $\delta W_e = \mathbf{Q}_e^T \delta \mathbf{q}$ , leads to the vector of generalized forces  $\mathbf{Q}_e$ . In order to include boundary conditions for a finite element or to connect finite elements, constraint equations are taken into account. These equations are conventionally included, utilizing a system of nonlinear algebraic equations, which depends on the generalized coordinates,

$$\mathbf{C}(\mathbf{q}, t) = 0 \quad (28)$$

in which  $\mathbf{C}$  is the vector of linearly independent constraint functions. This leads to an extension of the overall system of equations according to the number of finite elements. These additional equations could be avoided e.g. by a coordinate transformation to minimal coordinates. This would lead to a set of equations, where the degrees of freedom, which are relevant for e.g. robotics or control design, are directly available. However, the derivations to get this set of equations have to be performed symbolically for most of the applications. Furthermore, in the present contribution a sparse solver is used within the numerical time integration and therefore no significant computational costs result from the additional constraint equations. On the other hand, the present method provides a more general formulation of the overall system.

As an example for constraint equations, the functions  $\mathbf{C}_1$  to constrain the positions, and the function  $\mathbf{C}_2$  to constrain the angles of the elements  $i$  and  $j$  are shown in Eq. (29),

$$\begin{pmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{r}^i(x^i = L^i, t) - \mathbf{r}^j(x^j = 0, t) \\ \varphi^i - \frac{\partial w^i(x^i, t)}{\partial x^i} \Big|_{x^i = L^i} - \varphi^j + \frac{\partial w^j(x^j, t)}{\partial x^j} \Big|_{x^j = 0} \end{pmatrix} \quad (29)$$

Together with the constraint equations, see Eq. (28), the Lagrange equations, see Eq. (18), follow with

$$\begin{aligned} \mathbf{M} \ddot{\mathbf{q}} + \mathbf{K} \mathbf{q} + \mathbf{C}_q^T \lambda &= \mathbf{Q}_e^+ \mathbf{Q}_v \\ \mathbf{C}(\mathbf{q}, t) &= \mathbf{0} \end{aligned} \quad (30)$$

In Eq. (30)  $\mathbf{C}_q$  denotes the Jacobian of the constraint equations.

### 3 Absolute Nodal Coordinate Formulation

In the following, the planar ANCF element as derived by Berzeri and Shabana [15] is shortly revised. Note that symbols are introduced in this section which previously had a different meaning. The ANCF element has only eight degrees of freedom (DOF), four in each of the two nodes. Figure 3 shows the undeformed and the deformed configuration of the ANCF element. The vector  $\mathbf{r}$  denotes the actual position of a point on the deformed beam, which originally was located at the position  $x$  on the undeformed beam element. The nodal coordinates consist of two position DOF and two DOF for the directional derivative with respect to the  $x$  coordinate, also denoted as  $x$ -slope. The derivatives are abbreviated as  $\mathbf{r}' = \partial \mathbf{r} / \partial x$  and  $\mathbf{r}'' = \partial^2 \mathbf{r} / \partial x^2$ .

The advantage of this formulation compared to fully parameterized elements [16], which use all components of the gradient at a node, is the good convergence and small computational costs. The neglect of shear deformation is applicable in very thin and cable-like structures, which often appear in multibody systems.

The nodal coordinates at  $x = 0$  and  $x = L$  of an element are defined as

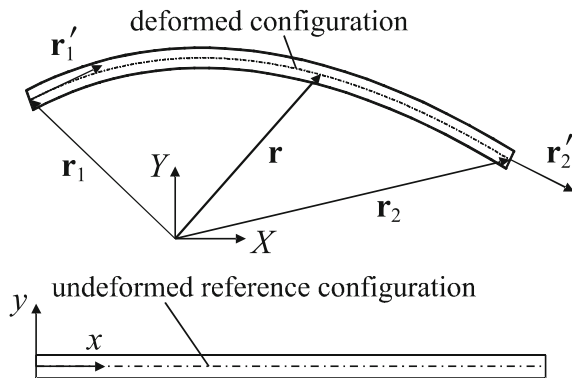
$$\mathbf{q}_j = \begin{bmatrix} \mathbf{r}_j^T & \mathbf{r}'_j^T \end{bmatrix}^T \quad (31)$$

and the element coordinates are

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_1^T & \mathbf{q}_2^T \end{bmatrix}^T \quad (32)$$

The actual position of the point  $\mathbf{r}$  is given by

$$\mathbf{r} = \begin{bmatrix} a_0 + a_1x + a_2x^2 + a_3x^3 \\ b_0 + b_1x + b_2x^2 + b_3x^3 \end{bmatrix} \quad (33)$$



**Fig. 3** ANCF model of a finite element

Note that this displacement field is a cubic polynomial in  $x$ . The coefficients  $a_i$  and  $b_i$  are chosen such that the displacement field fulfills the requirements of Eqs. (31) and (32) at the nodes, at position  $x = 0$  and  $x = L$ . This leads to

$$\mathbf{r} = [S_1 \mathbf{I} \quad S_2 \mathbf{I} \quad S_3 \mathbf{I} \quad S_4 \mathbf{I}] \mathbf{q} = \mathbf{S} \mathbf{q}. \quad (34)$$

The shape functions are given by

$$\begin{aligned} S_1 &= \frac{1}{2} - \frac{3}{4}\xi + \frac{\xi^3}{4}, \quad S_2 = \frac{L}{8} (1 - \xi - \xi^2 + \xi^3), \\ S_3 &= \frac{1}{2} + \frac{3}{4}\xi - \frac{\xi^3}{4}, \quad S_4 = \frac{L}{8} (-1 - \xi + \xi^2 + \xi^3) \end{aligned} \quad (35)$$

with the normalization  $\xi = 2x/L - 1$ . A comparison of the shape functions of the ANCF element of Eq. (35) and those of the FFRF element of Eqs. (10) and (11) shows that within the ANCF element for both, the axial and the transverse deflection the same polynomial is used. This leads to a total number of eight degrees of freedom, three of which take into account rigid body translations. Two shape functions describe the transverse deflection including the rotation at the boundary. The remaining three shape functions consequently take into account axial deformations, which therefore are approximated with cubic order. In comparison to the FFRF element, which takes into account axial deformation with a linear order of approximation, the equations of the ANCF element become much stiffer. However, within the FFRF element the transverse deflections are approximated with  $s + 2$  shape functions, where  $s \in [1 \dots 4]$ , such that the system of equations of the FFRF element becomes larger.

The mass matrix can be computed via integration over the  $x$ -axis, rewriting the shape functions with respect to the  $x$ -coordinate,

$$\mathbf{M} = A \int_0^L \rho \mathbf{S}^T \mathbf{S} dx \quad (36)$$

in which  $A$  denotes the cross-section area,  $L$  is the length and  $\rho$  is the density of the beam element. The mass matrix is integrated exactly by means of Gaussian quadrature. The computation of the elastic forces, which result from the virtual work of internal forces, is split into a part due to bending, based on the curvature of the deformed beam axis and a part due to axial strain. Usually in the ANCF, the axial strain is computed using the  $x$ -component of the Green strain tensor,

$$\varepsilon_{xx}^0 = \frac{1}{2} (\mathbf{r}'^T \mathbf{r}' - 1) \quad (37)$$



Due to compatibility reasons with classical finite beam elements, which are based on the geometrically exact solution of the beam centreline, the so called extensible elastica, or the shear deformable element provided by Simo and Vu-Quoc, in the present case a strain based on the axial stretch,

$$\varepsilon_0 = |\mathbf{r}'| - 1, \quad (38)$$

is used, compare e.g. the work of Dmitrochenko and Pogorelov [17].

The curvature of the beam centerline, which can be related to the curvature defined in the Frenet-frame, follows by

$$\kappa = \frac{|\mathbf{r}' \times \mathbf{r}''|}{|\mathbf{r}'|^3} \quad (39)$$

where  $\kappa$  is the absolute value of the curvature depending on a generalized parameter  $x$ . Using the Bernoulli–Euler beam theory, the virtual work of the elastic forces including axial deformation can be written as

$$W = \frac{1}{2} \int_0^L (EIK^2 + EA\varepsilon_0^2) dx \quad (40)$$

where  $K$  is the rate of rotation of the cross section with respect to the undeformed beam length, sometimes denoted as the “material form of curvature”, compare Reissner [18] and Gerstmayr and Matikainen [8]. This parameter is related to the rotation of the cross section  $\phi$ , respectively the spatial form of curvature  $\kappa$  by

$$K = \frac{\partial \phi}{\partial x} = \frac{\kappa}{|\mathbf{r}'|} \quad (41)$$

and leads to the exact solution for large deformation problems according to extensible elastica theories. In the implementation, it is possible to derive the integrand of Eq. (40) analytically in order to minimize the number of mathematical operations. The non-rational expressions of the integral in Eq. (40) can be well approximated by means of Gaussian integration. It turned out that five integration points for the axial strain components and three integration points for the curvature (bending) components are sufficient for accurate approximations of the elastic forces. The elastic forces can be derived from Eq. (40) by means of differentiation,

$$\mathbf{F}_{\text{el}}^T = \frac{\partial W}{\partial \mathbf{q}} = \int_0^L \left( EIK \frac{\partial K}{\partial \mathbf{q}} + EA\varepsilon_0 \frac{\partial \varepsilon_0}{\partial \mathbf{q}} \right) dx \quad (42)$$

The equations of motion are gained from D’Alembert’s principle and are written for one element

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{F}_{\text{el}}(\mathbf{q}) - \mathbf{C}_{,\mathbf{q}}^T \lambda = \mathbf{F}_{\text{ext}} \quad (43)$$

in which external forces  $\mathbf{F}_{\text{ext}}$  and the Jacobian  $\mathbf{C}_{,q}$  of the vector of constraints  $\mathbf{C}$  are considered. Compared to the equations of motion of one finite element described by the FFR formulation in Eq. (30) of the previous section, a constant mass-matrix is derived and centrifugal and Coriolis terms do not come into the play. However, the vector of elastic forces is nonlinear in this case, which leads to a nonlinear stiffness matrix.

## 4 Numerical Examples

In the following, large deformation static and dynamic example problems are presented. The investigated mechanical systems are modeled with ANCF and FFRF beam finite elements for comparison. In case of static computations, the convergence rate is investigated and the results are compared to an analytical solution of the extensible Euler elastica, which is provided according to the work of Gerstmayr and Irschik [9].

In case of dynamic computations, a reference value based on the strain energy of the finite elements is provided. This reference value allows a simplified comparison of different formulations and can act as reference solution for future studies.

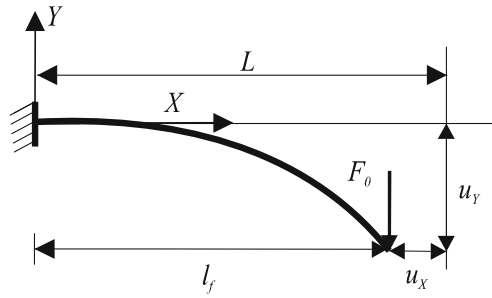
All computations are performed within the multibody code HOTINT, where accuracy parameters and criteria for the convergence of the numerical differentiation were set, such that highest accuracy could be reached for the investigation of the convergence.

Characteristic solutions of both, static and dynamic examples, are shown at the end of each section.

### 4.1 Static Example Problem

In the following static example problem, a cantilever beam with a tip load is imposed. The length of the beam is chosen to be  $L = 2m$ , the rectangular cross-section is given by  $h = b = 0.01m$  and Young's Modulus is  $E = 2.07 \times 10^{11} \text{N/m}^2$ . The tip load is chosen  $F_0 = 1.5EI/L^2$ , such that large deformations occur, see Fig. 4.

The example problem is modeled by means of ANCF and FFRF beam finite elements. For comparison, the tip displacement is calculated with an increasing number of finite elements. The results are reported for a comparison of convergence of the method itself. When convergence is reached the solutions are compared additionally with the analytical solution of the extensible Euler elastica as proposed in the work of Gerstmayr and Irschik [9]. Following this contribution, the differential equations of the cantilever beam as shown in Fig. 4 are derived from the local equilibrium equations of a stretched beam element

**Fig. 4** Cantilever beam

$$\frac{\partial^2 \theta(x)}{\partial x^2} = -(1 + \lambda(x)) \frac{F_0}{EI} \cos(\theta(x)). \quad (44)$$

where  $\theta$  denotes the actual rotation of the beam axis with respect to the undeformed beam coordinate.  $EI$  includes the bending stiffness and  $\lambda$  the axial elongation given as

$$\lambda(x) = \frac{F_0}{EA} \sin(\theta(x)). \quad (45)$$

The boundary conditions for the cantilever beam read

$$\theta(x)|_{x=0} = 0 \quad \text{and} \quad \left. \frac{\partial \theta(x)}{\partial x} \right|_{x=0} = \frac{F_0 l_F}{EI}, \quad (46)$$

meaning that the rotation at the support is zero and the bending moment at the support is  $M = F_0 l_F$ . The horizontal distance  $l_F$  between the force  $F_0$  and the support in deformed configuration, as shown in Fig. 4, depends on the unknown solution for the rotation, but needs to be taken into account for large deformation analysis. The distance can be computed by

$$l_F = \int_0^L [(\lambda + 1) \cos(\theta(x)) + 1] dx \quad (47)$$

and needs to be solved together with Eq. (44). The nonlinear deformation problem can be solved for arbitrary precision in symbolic computer programs by means of an iterative solution of  $l_F$ . Initially, the distance of the force  $F_0$  is assumed to be  $l_F = L$ , and in the  $i$ th iteration, the solution of the last iteration  $\theta^{(i-1)}(s)$  is used to compute the approximated distance of the applied force, which converges well for the investigated problems. To calculate the reference solution, the method has been implemented in the symbolic computer code MAPLE, which in general provides arbitrary accuracy. For the following example, the tolerance of the Newton method has been set to  $10^{-18}$ , while 22 digits have been used internally for the computation of the single expressions in MAPLE.

**Table 1** Displacements in the cantilever beam

No. elements	ANCF		FFRF	
	$u_X(\text{m})$	$u_Y(\text{m})$	$u_X(\text{m})$	$u_Y(\text{m})$
64	0.2158812656065	0.8219585119628	0.2158740	0.8219675
128	0.2158812676722	0.8219585172209	0.2158794	0.8219607
256	0.2158812678120	0.8219585174868	0.2158808	0.8219590
512	0.2158812678139	0.8219585174889	0.2158811	0.8219586
Extensible elastica	0.2158812678134	0.8219585174888	0.2158812	0.8219585

**Table 2** Displacements in the cantilever beam

Elements	$u_X(\text{m})$	$u_Y(\text{m})$
4096 FFRF	0.2158812660572	0.8219585119628
Elastica	0.2158812678134	0.8219585174888

The distances  $u_X$  and  $u_Y$  of the endpoint with respect to the  $(X-, Y-)$  coordinate system are shown for an increasing number of beam finite elements in Table 1. There, a faster convergence of the ANCF beam finite element, which coincides up to 12 digits with the solution of the elastica when taking into account 512 beam finite elements, is shown.

The results of the FFRF model show a comparable tendency towards the same reference value, given by the solution of the elastica. Therefore an additional calculation taking into account 4096 FFRF beam finite elements is performed. The results of this calculation are shown in Table 2, where a significant increase of coinciding digits is documented.

The convergence of both formulations is shown in Figs. 5 and 6 by means of the relative errors of the  $X$ - and  $Y$ -displacements, given for a quantity  $u$  and a corresponding reference value  $u_{ref}$  by

$$err_u = \frac{|u - u_{ref}|}{|u_{ref}|}, \quad (48)$$

where the corresponding solution of the elastica is used as a reference value. On the logarithmic scale, the order of convergence for the FFRF elements is approximately  $O(n^2)$ , for both the  $X$ - and the  $Y$ -displacements. The ANCF elements converge approximately with  $O(n^2)$  for a lower number of finite elements, while the convergence rises significantly for a larger number of finite elements to nearly  $O(n^6)$ .

## 4.2 Dynamic Example Problems

In the following section a slider crank mechanism is treated exemplarily for comparison of the ANCF and the FFRF within a dynamic example. The mechanism consists of a rigid crank of length  $r = 0.1$  m, which is driven by a constant circular velocity  $\omega = 25$  rad/s. The flexible slider has length  $L = 0.5$  m, cross-sectional

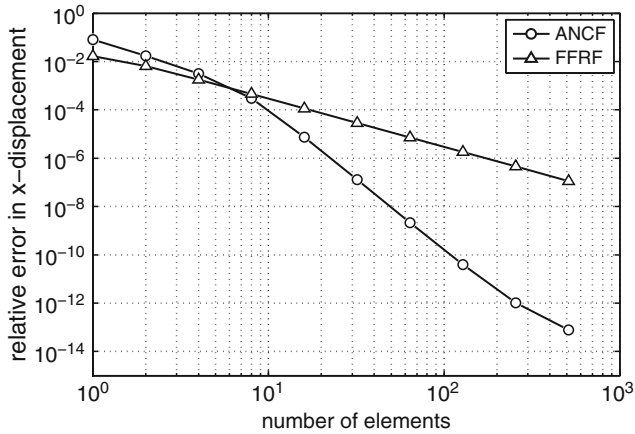


Fig. 5 Relative error of the tip displacements in  $X$ -direction versus the number of elements

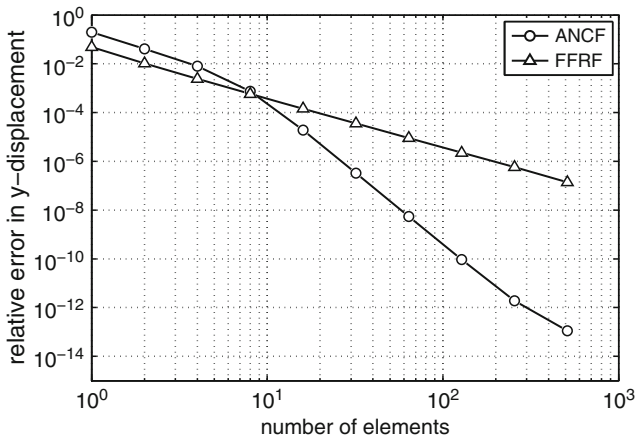


Fig. 6 Relative error of the tip displacements in  $Y$ -direction versus the number of elements

area  $A = hb = 0.004 \times 0.004 \text{ m}^2$  and gravity  $g = 9.81 \text{ m/s}^2$ . In order to provide a highly flexible system, Young's modulus is reduced to  $E = 2.07 \times 10^{10} \text{ N/m}^2$ . In case of the FFRF model  $s = 2$  eigenmodes are chosen. The connection rod is constrained to the prescribed motion of the crank at the left end, and cannot move in vertical direction at the right end, compare Fig. 7. The slider has an initially undeformed straight position and is aligned in the global  $X$ -direction together with the rigid crank. The initial velocity of the slider is zero.

The prescribed velocity at the hinge, therefore leads to high (but limited) accelerations, which induce large oscillations. The  $X$ -position of the right end and the  $Y$ -position of the midpoint of the slider are presented in Figs. 8 and 9, respectively. The comparison of eight and 64 finite elements shows already a good overall convergence of the models.

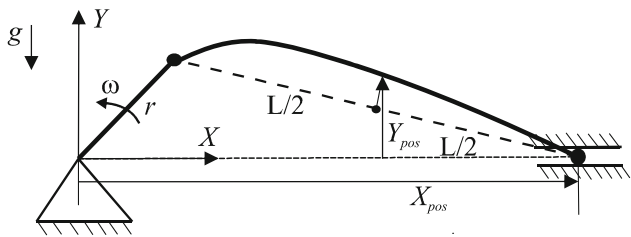


Fig. 7 Flexible slider-crank mechanism, deformed configuration

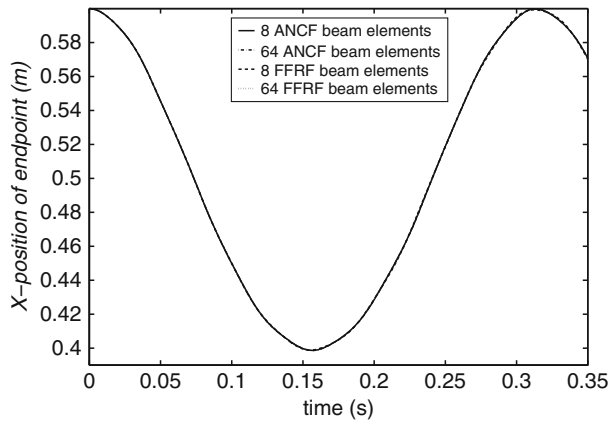


Fig. 8 Time evolution of the X-position of the right end of the slider

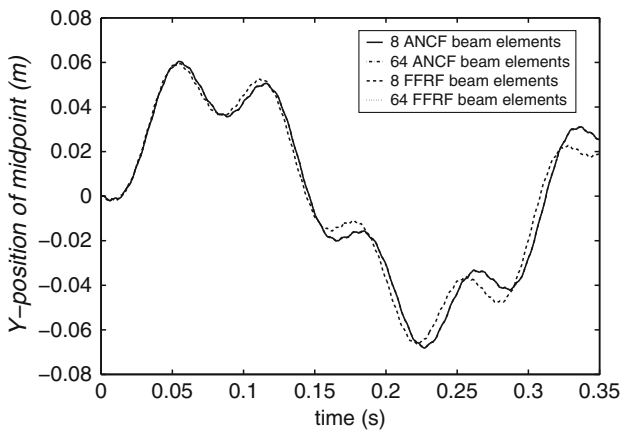


Fig. 9 Time evolution of the Y-position of the mid-point of the slider

In order to simplify the comparison of convergence of dynamic examples, one characteristic reference value, comparable to the endpoint deflection of a static cantilever beam, is provided in the following. This value is supposed to represent elastic effects of the studied system. Therefore, the strain energy of the beam finite elements,

$$U = \frac{1}{2} \int_V (\boldsymbol{\sigma}^T \boldsymbol{\epsilon}) dV = \frac{1}{2} \left[ EA \int_L (u')^2 dx + EI \int_L (w'')^2 dx \right], \quad (49)$$

is used as a basis. In order not to eliminate any shares of the strain energy, due to alternating amplitudes, the absolute value is taken into account. Finally, to gain the reference value  $R$ , the arithmetic mean value of the studied time period  $t$  is imposed such that

$$R = \frac{1}{t} \int_t |U| d\tau. \quad (50)$$

The reference values are shown for both, an ANCF and a FFRF model of the slider-crank mechanism in Table 3, where a tendency towards a converged solution can be assumed.

This convergence tendency of the problem example with respect to the method itself, consequently is documented by means of the error value

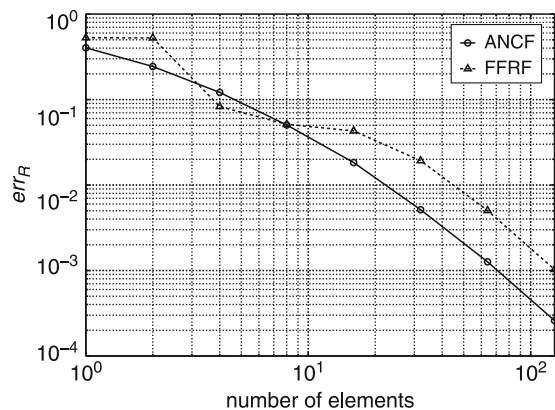
$$err_R = \frac{|R - R_{ref}|}{R_{ref}}, \quad (51)$$

where  $R_{ref}$  is the value of a reference solution given by 256 beam finite elements in the present case. The corresponding convergence behavior of the slider-crank mechanism is shown in Fig. 10. Again, the ANCF model shows a better convergence behavior, but both formulations lead to the same results in the converged case.

**Table 3** Reference value of ANCF and FFRF slider-crank model

No. elements	$R_{\text{ANCF}}$	$R_{\text{FFRF}}$
1	0.020907904853330057	0.053579719281086631
2	0.026523065528398459	0.053494855207199613
4	0.030813768896922035	0.032150639794887304
8	0.033293322837706731	0.033255123185251721
16	0.034418309164295127	0.033545293772669946
32	0.034880005393381937	0.034377494644696992
64	0.035015858976338687	0.034875805279761446
128	0.035051064338982495	0.035016179848660837
256	0.035060163996004430	0.035052237300640429

**Fig. 10** Error value with respect to strain energy; slider-crank mechanism



## 5 Conclusions

In the present contribution, the fundamental different approaches ANCF and FFRF for modeling and simulation of multibody systems including large deformations have been compared. Although the methods use a completely different discretization, they lead all to the same results in the converged case. In the static case a cantilever beam with a tip load was treated, where the endpoint deflection was used as a reference value to study the convergence behavior of each method with respect to the analytical solution of the extensible Euler elastica. A slider-crank mechanism was used exemplarily to study a dynamic example. In order to enable a simplified comparison of the results of dynamic examples, a reference value based on the strain energy of beam finite elements was introduced. The proposed entity may act as a reference value in future studies.

It is noteworthy that in both examples the ANCF model showed better convergence properties.

**Acknowledgement** Support of the present work in the framework of the COMET K2 Austrian Center of Competence in Mechatronics (ACCM) is gratefully acknowledged.

## References

1. Ashley H (1967) Observation of the dynamic behavior of flexible bodies in orbit. *AIAA J* 5(3):460–469
2. De Veubeke BJ (1976) The dynamics of flexible bodies. *Int J Eng Sci* 14:895–913
3. Bathe KJ (1990) *Finite-elemente-methoden*. Springer, Berlin/Heidelberg/New York
4. Rodríguez JI, Jiménez JM, Funes FM, García de Jalon J (2004) Recursive and residual algorithms for the efficient numerical integration of multi-body systems, *Multibody Syst Dyn* 11:295–320
5. Simo JC, Vu-Quoc L (1986) On the dynamics of flexible beams under large overall motions – the plane case: parts I and II. *J Appl Mech* 53:849–863



6. Irschik H, Gerstmayr J (2009) A continuum mechanics based derivation of Reissner's large-displacement finite-strain beam theory: the case of plane deformations of originally straight Bernoulli-Euler beams. *Acta Mech* 206:1–21
7. Shabana AA, Hussien H, Escalona J (1998) Application of the absolute nodal coordinate formulation to large rotation and large deformation problems. *ASME J Mech Design* 120:188–195
8. Gerstmayr J, Matikainen MK (2006) Improvement of the accuracy of stress and strain in the absolute nodal coordinate formulation. *Mech Based Design Struct Mach* 34(4):409–430
9. Gerstmayr J, Irschik H (2008) On the correct representation of bending and axial deformation in the absolute nodal coordinate formulation with an elastic line approach. *J Sound Vibrat* 318:461–487
10. Dibold M, Gerstmayr J, Irschik H (2009) Verification of structural elements in large deformation problems. CD-Proceedings of ECCOMAS Thematic Conference: Multibody Dynamics 2009, Warsaw, Poland, 29 June–2 July 2009
11. Shabana AA (1998) *Dynamics of multibody systems*, 2nd edn. Cambridge University Press, Cambridge
12. Hurty WC (1965) Dynamic analysis of structural systems using component modes. *AIAA J* 3(4):678–685
13. Dibold M, Gerstmayr J, Irschik H (2009) A detailed comparison of the absolute nodal coordinate and the floating frame of reference formulation in deformable multibody systems. *J Nonlinear Comput Dyn* 4(2):021006–1–021006–10
14. Ziegler F (1991) *Mechanics of solids and fluids*. Springer, New York
15. Berzeri M, Shabana AA (2000) Development of simple models for the elastic forces in the absolute nodal co-ordinate formulation. *J Sound Vibrat* 235(4):539–565
16. Yakoub RY, Shabana AA (2001) Three dimensional absolute nodal coordinate formulation for beam elements: implementation and application. *J Mech Design* 123:614–621
17. Dmitrochenko ON, Pogorelov DY (2003) Generalization of plate finite elements for absolute nodal coordinate formulation. *Multibody Syst Dyn* 10:17–43
18. Reissner E (1972) On one-dimensional finite-strain beam theory: the plane problem. *Zeitschrift für Angew Math Phys* 23:795–804

# Modeling and Analysis of Rigid Multibody Systems with Translational Clearance Joints Based on the Nonsmooth Dynamics Approach

Paulo Flores, Remco Leine, and Christoph Glocker

**Abstract** The main purpose of this paper is to discuss a method for a dynamic modeling and analysis of multibody systems with translational clearance joints. The method is based on the nonsmooth dynamics approach, in which the interaction of the elements that constitute a translational clearance joint is modeled with multiple frictional unilateral constraints. In the following, the most fundamental issues of the nonsmooth dynamics theory are revised. The dynamics of rigid multibody systems are stated as an equality of measures, which are formulated at the velocity-impulse level. The equations of motion are complemented with constitutive laws for the normal and tangential directions. In this work, the unilateral constraints are described by a set-valued force law of the type of Signorini's condition, while the frictional contacts are characterized by a set-valued force law of the type of Coulomb's law for dry friction. The resulting contact-impact problem is formulated and solved as a linear complementarity problem, which is embedded in the Moreau's time-stepping method. Finally, the classical slider-crank mechanism is considered as a demonstrative application example and numerical results are presented. The obtained results show that the existence of clearance joints in the modeling of multibody systems influences their dynamics response.

## 1 Introduction

Manufacturing tolerances, wear and material deformation lead to imperfect joints and, therefore, clearances. These clearances modify the dynamic response of the system, justify the deviations between the numerical predictions and the

---

P. Flores (✉)

Mechanical Engineering Department, University of Minho, Campus de Azurém,  
4800-058 Guimarães, Portugal  
e-mail: [pflores@dem.uminho.pt](mailto:pflores@dem.uminho.pt)

R. Leine and C. Glocker

Center of Mechanics, IMES – Institute of Mechanical Systems, Department  
of Mechanical and Process Engineering, CH-8092 Zurich, Switzerland  
e-mail: [remco.leine@imes.mavt.ethz.ch](mailto:remco.leine@imes.mavt.ethz.ch); [christoph.glocker@imes.mavt.ethz.ch](mailto:christoph.glocker@imes.mavt.ethz.ch)

experimental measurements and eventually lead to important deviations between the projected behavior of the mechanisms and their real outcome. The presence of clearance in joints is a complex and important issue in the realistic modeling of multibody systems. This aspect gains paramount importance due to the demand for the proper design of the real joints in many industrial applications. Over the last few years, extensive work has been done to study the dynamic effect of the revolute joints with clearance in multibody systems. However, translational joints with clearance have received less attention [1–4].

Indeed, a number of theoretical and experimental works devoted to the research on multibody mechanical systems with realistic joints has been published recently. However, most of these works focus on revolute joints with and without lubrication effects. An extensive literature review on the issue of modeling and simulation of multibody systems with revolute and spherical clearance joints can be found in the work by Flores et al. [2]. In contrast to the revolute and spherical clearance joints, not much work has been done to model translational joints with clearance because in this case several different configurations between the joints elements can take place. In fact, the contact configurations of slider and guide include: (i) no contact between the two elements; (ii) one corner of the slider is in contact with the guide surface; (iii) two adjacent slider corners are in contact with the guide surface, which corresponds to have a face of the slider in contact with the guide surface; (iv) two opposite slider corners are in contact with the guide surface [5–7]. Moreover, each contact point may be in stick or in slip phase, which greatly enlarges the number of contact configuration. The conditions for switching from one case to another depend on the system's dynamic response.

Farahanchi and Shaw [8] studied the dynamic response of a planar slider-crank mechanism with slider clearance. They demonstrated how complex the system's response is, which can be chaotic or periodic. More recently, Thümmel and Funk [9] used the complementarity approach to model impact and friction in a slider-crank mechanism with both revolute and translational clearance joints. With the purpose to analyze the slider crank mechanism, Wilson and Fawcett [10] derived the equations of motion for all different possible configurations of the slider motion inside the guide, resulting in a total of 40 equations. They also showed how the slider motion in a translational clearance joint depends on the geometry, speed and mass distribution.

Therefore, in the present work, the nonsmooth dynamics approach is used to model the type of multibody systems, due to its simplicity and ability to deal with all possible different configurations in a unified manner. The methodology is based on the nonsmooth dynamics approach, in which the interaction of the colliding bodies is modeled with multiple frictional unilateral constraints. The dynamics of rigid multibody systems are stated as an equality of measures, which are formulated at the velocity-impulse level. The equations of motion are complemented with constitutive laws for the forces and impulses the normal and tangential directions. In this work, the unilateral constraints are described by a set-valued force law of the type of Signorini's condition, while the frictional contacts are characterized by

a set-valued force law of the type of Coulomb's law for dry friction. The resulting contact-impact problem is formulated and solved as a linear complementarity problem, which is embedded in the Moreau's time-stepping method

## 2 Basic Set-Valued Elements

A linear complementarity problem (LCP) is a set of linear equations that can be written as, [11, 12]

$$\mathbf{y} = \mathbf{Ax} + \mathbf{b} \quad (1)$$

subjected to the inequality complementarity conditions

$$\mathbf{y} \geq \mathbf{0}, \quad \mathbf{x} \geq \mathbf{0}, \quad \mathbf{y}^T \mathbf{x} = 0 \quad (2)$$

for which the vectors  $\mathbf{x}$  and  $\mathbf{y}$  have to be evaluated for given  $\mathbf{A}$  and  $\mathbf{b}$ . In other words, the LCP is the problem of finding solutions  $\mathbf{x} \in \mathbb{R}^n$  and  $\mathbf{y} \in \mathbb{R}^n$  of (1) and (2), where  $\mathbf{b}$  is an  $n$ -dimensional constant column and  $\mathbf{A}$  is a given square matrix of dimension  $n$ . The inequality complementarity conditions expressed by Eq. (2) are often written in the form

$$\mathbf{0} \leq \mathbf{y} \perp \mathbf{x} \geq \mathbf{0} \quad (3)$$

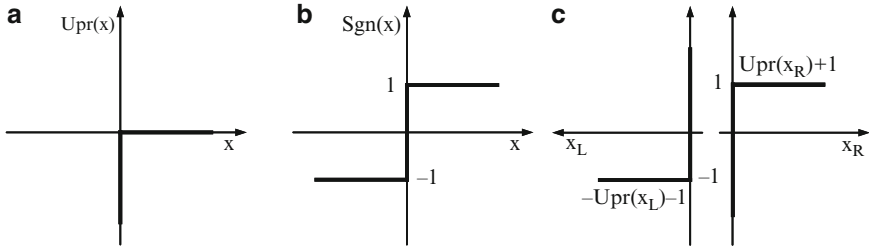
where  $\mathbf{y} \perp \mathbf{x}$  denotes  $\mathbf{y}^T \mathbf{x} = 0$ . An LCP can have a unique solution, multiple solutions or no solution at all [13, 14]. All existing solutions can be found using enumerative methods, which treat the problem by a combinatorial evolution of the complementarity condition  $x_i y_i = 0$ . From the complementarity condition it follows that when  $x_i > 0$ , then  $y_i = 0$ , and vice versa. An LCP of dimension  $n$  provides  $2^n$  different combinations of  $n$  variables, which are allowed to be greater than zero at the same time. For large dimensions, enumerative methods become numerically expensive since  $2^n$  grows rapidly. A more efficient algorithm is the complementarity pivot algorithm, usually referred to as Lemke's algorithm [15–17]. Other efficient algorithms to solve LCP can be found in the work by Cottle et al. [12].

One of the most important multifunctions (or set-valued maps) related to complementarity is the unilateral primitive, denoted by  $\text{Upr}$ . The unilateral primitive is a maximal monotone set-valued map on  $\mathbb{R}_0^+$  defined as [18, 19]

$$\text{Upr}(x) := \begin{cases} \{0\} & x > 0 \\ (-\infty, 0] & x = 0 \\ \emptyset & x < 0 \end{cases} \quad (4)$$

The graph of the unilateral primitive map is depicted in Fig. 1a. Thus, each complementarity condition of an LCP can be expressed as one  $\text{Upr}$  inclusion

$$-y \in \text{Upr}(x) \quad \Leftrightarrow \quad y \geq 0, \quad x \geq 0, \quad xy = 0 \quad (5)$$



**Fig. 1** (a) Map  $x \rightarrow \text{Upr}(x)$ ; (b) Map  $x \rightarrow \text{Sgn}(x)$ ; (c) decomposition  $\text{Sgn}(x)$  into  $\text{Upr}(x)$

Unilateral primitives are used in mechanics at the displacement level and at the velocity level to model unilateral geometric and kinematic constraints, such as free plays with stops, sprag clutches among others. The associated set-valued force laws are conveniently stated as inclusions of (5).

A second maximal monotone set-valued map, frequently used in complementarity problems, is the filled-in relay function  $\text{Sgn}$ -multifunction, which is defined by [18, 19]

$$\text{Sgn}(x) := \begin{cases} \{+1\} & x > 0 \\ [-1, +1] & x = 0 \\ \{-1\} & x < 0 \end{cases} \quad (6)$$

It is important to highlight that, while the classical  $\text{sgn}$ -function is defined with  $\text{sgn}(0) = 0$ , the  $\text{Sgn}$ -multifunction is set-valued at  $x = 0$ . The graph of the  $\text{Sgn}$ -multifunction is shown in Fig. 1b. An inclusion in the  $\text{Sgn}$ -multifunction can always be represented by two inclusions involving the unilateral primitive. The decomposition can be written as

$$-y \in \text{Sgn}(x) \Leftrightarrow \exists x_R, x_L \text{ s.t. } \begin{cases} -y \in +\text{Upr}(x_R) + 1 \\ -y \in -\text{Upr}(x_L) - 1 \\ x = x_R - x_L \end{cases} \quad (7)$$

Using Eq. (5), the Eq. (7) can be rewritten in terms of complementarities

$$-y \in \text{Sgn}(x) \Leftrightarrow \exists x_R, x_L \text{ s.t. } \begin{cases} 1 + y \geq 0, & x_R \geq 0, & (1 + y)x_R = 0 \\ 1 - y \geq 0, & x_L \geq 0, & (1 - y)x_L = 0 \\ x = x_R - x_L \end{cases} \quad (8)$$

This representation has to be used when a problem involving  $\text{Sgn}$ -multifunctions is formulated as an LCP in its standard form [20].

### 3 Set-Valued Force Laws for Frictional Unilateral Contacts

In the present work, the normal contact between rigid bodies is characterized by a set-valued force law called Signorini's condition [21]. Figure 2 shows two convex rigid bodies apart from each other by a relative normal gap or distance denoted by  $g_N$ . This relative normal gap is uniquely defined for convex surfaces, being perpendicular to the tangent planes at the contact points 1 and 2. The relative normal gap is non-negative due to impenetrability condition of the bodies. The two bodies in contact with each other when  $g_N = 0$ . In fact, one of the main features of unilateral contact is the impenetrability condition, which means that the candidate bodies for contact must not cross the boundaries of antagonist bodies. On the other hand, the normal contact force  $\lambda_N$  is also non-negative because the bodies can not attract each other, that is, the constraint is unilateral. The normal contact force vanishes when there is no contact, i.e.,  $g_N > 0$ , and can only be positive when contact happens, that is,  $g_N = 0$ . Thus, under the assumption of impenetrability between the bodies, expressed by  $g_N \geq 0$ , only two situations can occur, namely.

$$g_N = 0 \wedge \lambda_N \geq 0, \quad (\text{closed contact}) \quad (9)$$

$$g_N > 0 \wedge \lambda_N = 0, \quad (\text{open contact}) \quad (10)$$

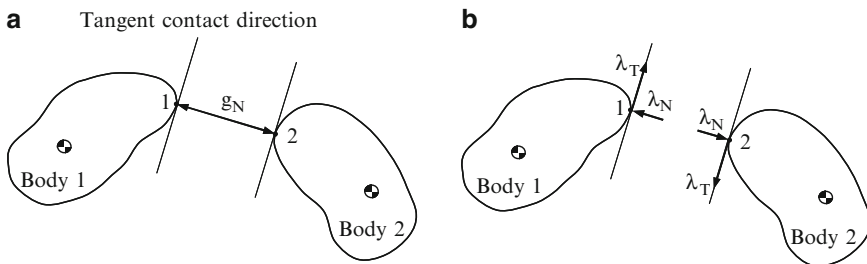
Equations (9) and (10) represent an inequality complementarity behavior, for which the product of the relative normal gap and normal contact force is zero; i.e.

$$g_N \lambda_N = 0 \quad (11)$$

The relation between the normal gap and normal contact force is described by

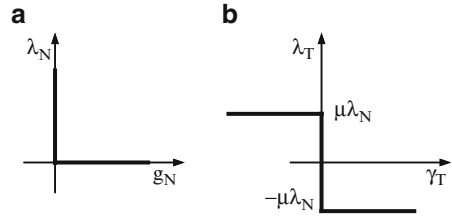
$$g_N \geq 0, \quad \lambda_N \geq 0, \quad g_N \lambda_N = 0 \quad (12)$$

which represents the inequality complementarity condition between  $g_N$  and  $\lambda_N$ , the so-called Signorini's condition.



**Fig. 2** (a) Relative normal gap; (b) normal and tangential contact forces

**Fig. 3** (a) Signorini's normal contact law; (b) Coulomb's friction law



The inequality complementarity behavior of the normal contact law is depicted in Fig. 3a and shows a set-valued graph or a corner of admissible combinations between  $g_N$  and  $\lambda_N$  [22]. When two rigid bodies are contacting, the Signorini's condition given by Eq. (12) needs to be complemented with an impact law, such as the well known Newton's kinematical law that relates the pre- and post-impact velocities to the bodies' normal coefficient of restitution,  $\varepsilon_N$ .

The classical Coulomb's friction law is another typical example that can be considered as a set-valued force law [18, 23–25]. The magnitude of the static friction force is less than or equal to the maximum static friction force which is also proportional to the normal contact force. Consider again the two contacting rigid bodies depicted in Fig. 2, in which Coulomb friction is present at the contact points 1 and 2. The relative velocity of point 1 with respect to point 2 along their tangent plane is denoted by  $\gamma_T$ . If contact between the two bodies takes place, i.e.  $g_N = 0$ , then the friction phenomenon imposes a tangential force  $\lambda_T$  as is illustrated in Fig. 2b. If the bodies are sliding over each other, the friction force  $\lambda_T$  has the magnitude  $\mu \lambda_N$  and acts in the direction opposed to the relative tangential velocity

$$-\lambda_T = \mu \lambda_N \text{Sgn}(\gamma_T) \quad \gamma_T \neq 0 \quad (13)$$

where  $\mu$  is the friction coefficient and  $\lambda_N$  is the normal contact force. If the relative tangential velocity vanishes,  $\gamma_T = 0$ , then the bodies purely roll over each other without slip. Pure rolling, or no-slip for locally flat objects, is denoted by stick. If the bodies stick, then the friction force must lie in the interval  $-\mu \lambda_N \leq \lambda_T \leq \mu \lambda_N$ . For unidirectional friction three different scenarios can occur, namely

$$\gamma_T = 0 \Rightarrow |\lambda_T| \leq \mu \lambda_N \quad (\text{sticking}) \quad (14)$$

$$\gamma_T < 0 \Rightarrow \lambda_T = +\mu \lambda_N \quad (\text{negative sliding}) \quad (15)$$

$$\gamma_T > 0 \Rightarrow \lambda_T = -\mu \lambda_N \quad (\text{positive sliding}) \quad (16)$$

These three scenarios can be summarized by a set-valued force law as

$$-\lambda_T \in \mu \lambda_N \text{Sgn}(\gamma_T) \quad (17)$$

Figure 3b shows the Coulomb's friction law as a set-valued force law [18].

## 4 Dynamics of Nonsmooth Rigid Multibody Systems

From classical mechanics, it is well known that the Newton–Euler equations of motion of a multibody system with  $f$  degrees of freedom and with only frictionless bilateral constraints can be written as [26]

$$\mathbf{M}\dot{\mathbf{u}} - \mathbf{h} = \mathbf{0} \quad (18)$$

$$\dot{\mathbf{q}} = \mathbf{u} \quad \forall t \quad (19)$$

where  $\mathbf{M} = \mathbf{M}(\mathbf{q}, t) \in \mathbb{R}^{f \times f}$  is the positive definite and symmetric mass matrix,  $\mathbf{h} = \mathbf{h}(\mathbf{q}, \mathbf{u}, t) \in \mathbb{R}^f$  represents the vector of all external and gyroscopic forces acting on the system forces originating from springs and dampers are also included in vector  $\mathbf{h}$ ,  $\mathbf{q} = \mathbf{q}(t) \in \mathbb{R}^f$  is the  $f$ -dimensional vector of generalized coordinates,  $\mathbf{u} = \mathbf{u}(t) \in \mathbb{R}^f$  addresses the system generalized velocities and  $\dot{\mathbf{u}} = \dot{\mathbf{u}}(t) \in \mathbb{R}^f$  is the vector that contains the system accelerations.

It is clear that Eq. (18) represents a classical second-order differential equation that describes the dynamic behavior of a multibody system without any contacts and contact forces. Therefore, when a system includes frictional unilateral constraints, the occurring contact forces should be taken into account in the equations of motion. In general, the magnitudes of the normal and tangential contact forces are added to the equations of motion by using the Lagrange multiplier technique [27]. Thus, adding the contact forces to Eq. (18), the dynamic equations of motion of a rigid multibody system with normal and tangential contact forces can, for non-impulsive motion, be written on the acceleration level as [19, 23]

$$\mathbf{M}\dot{\mathbf{u}} - \mathbf{h} - \mathbf{W}_N \boldsymbol{\lambda}_N - \mathbf{W}_T \boldsymbol{\lambda}_T = \mathbf{0} \text{ a.e.} \quad (20)$$

$$\dot{\mathbf{q}} = \mathbf{u} \quad \forall t \quad (21)$$

where  $\mathbf{W}_N = \mathbf{W}_N(\mathbf{q}, t) \in \mathbb{R}^{f \times n}$  and  $\mathbf{W}_T = \mathbf{W}_T(\mathbf{q}, t) \in \mathbb{R}^{f \times n}$  gather the generalized normal and tangential force directions  $\mathbf{w}_{Ni}$  and  $\mathbf{w}_{Ti}$ , respectively. The normal and tangential contact forces have magnitudes  $\lambda_{Ni}$  and  $\lambda_{Ti}$  for each contact point  $i$ . The dual variables to the normal contact forces  $\boldsymbol{\lambda}_N$  are the variations of normal gap distances  $\mathbf{g}_N$ , while the dual variables to the generalized friction or tangential forces  $\boldsymbol{\lambda}_T$  are the variations of the generalized sliding velocities  $\boldsymbol{\gamma}_T$ . The remaining terms of Eq. (20) have the same meaning as described above. It is important to note that Eq. (20) requires the existence of the velocities  $\mathbf{u}$  as well as the existence of accelerations  $\dot{\mathbf{u}}$ . Motion without impulses implies that  $\boldsymbol{\lambda}_N(t)$  is (locally) bounded and time-continuous. The velocities  $\mathbf{u}(t)$  therefore exist on non-impulsive time-intervals. The friction force  $\boldsymbol{\lambda}_T(t)$  is discontinuous when a slip-stick transition takes place or when the relative sliding velocity of a frictional contact reverses its sign. The acceleration  $\dot{\mathbf{u}}$  is not defined when  $\boldsymbol{\lambda}_T(t)$  is discontinuous. The set of time instances for which  $\boldsymbol{\lambda}_T(t)$  is discontinuous is of measure zero and Eq. (20), therefore, holds for almost all  $t$ .



Impulsive motion is described by the impact equation,

$$\mathbf{M}(\mathbf{u}^+ - \mathbf{u}^-) - \mathbf{W}_N \mathbf{\Lambda}_N - \mathbf{W}_T \mathbf{\Lambda}_T = \mathbf{0} \quad a.e. \quad (22)$$

$$\mathbf{u}^+(t) = \lim_{\Delta t \downarrow 0} \frac{\mathbf{q}(t + \Delta t) - \mathbf{q}(t)}{\Delta t}, \quad \mathbf{u}^-(t) = \lim_{\Delta t \uparrow 0} \frac{\mathbf{q}(t + \Delta t) - \mathbf{q}(t)}{\Delta t} \quad (23)$$

which relates the velocity jump to the impulsive forces  $\mathbf{\Lambda}_N$  and  $\mathbf{\Lambda}_T$  in normal and tangential direction respectively. We assume that the velocities  $\mathbf{u}(t)$  are of locally bounded variation (without singular part) and denote  $\mathbf{u}^-(t)$  and  $\mathbf{u}^+(t)$  as the pre- and post-impact velocity respectively. Furthermore, note that finite forces, such as gravity or reaction forces from springs and dampers, are non-impulsive, and do not occur in Eq. (22).

Following Moreau [28] we will cast the non-impulsive dynamics (20) and the impulsive dynamics (22) in a unified description, by using an equality of measures. This constitutes the general framework for nonsmooth rigid multibody dynamics [24, 29].

Multiplying the equation of motion (20) with the Lebesgue measure  $dt$  and the impact equation (22) with the atomic measure  $d\eta$ , being the sum of the Dirac point measures at the impact times, yields

$$\mathbf{M}\dot{\mathbf{u}}dt - \mathbf{h}dt - \mathbf{W}_N \boldsymbol{\lambda}_N dt - \mathbf{W}_T \boldsymbol{\lambda}_T dt = \mathbf{0} \quad (24)$$

$$\mathbf{M}(\mathbf{u}^+ - \mathbf{u}^-) d\eta - \mathbf{W}_N \mathbf{\Lambda}_N d\eta - \mathbf{W}_T \mathbf{\Lambda}_T d\eta = \mathbf{0} \quad (25)$$

Addition of Eqs. (24) and (25) results in

$$\mathbf{M}[\dot{\mathbf{u}}dt + (\mathbf{u}^+ - \mathbf{u}^-) d\eta] - \mathbf{h}dt - \mathbf{W}_N (\boldsymbol{\lambda}_N dt + \mathbf{\Lambda}_N d\eta) - \mathbf{W}_T (\boldsymbol{\lambda}_T dt + \mathbf{\Lambda}_T d\eta) = \mathbf{0} \quad (26)$$

or more briefly,

$$\mathbf{M}d\mathbf{u} - \mathbf{h}dt - \mathbf{W}_N d\mathbf{P}_N - \mathbf{W}_T d\mathbf{P}_T = \mathbf{0} \quad (27)$$

The differential measure for the velocities  $d\mathbf{u} = \dot{\mathbf{u}}dt + (\mathbf{u}^+ - \mathbf{u}^-)d\eta$  consists of the Lebesgue measurable part  $\dot{\mathbf{u}}dt$ , which accounts for absolutely continuous motion, and the atomic parts which accounts for impulsive motion. Hence, for impact free motion it holds that  $d\mathbf{u} = \dot{\mathbf{u}}dt$ . Similarly, the measure for the so-called percussions corresponds to a Lagrangian multiplier which gathers both finite contact forces  $\boldsymbol{\lambda}$  and impulsive contact forces  $\mathbf{\Lambda}$ , that is,  $d\mathbf{P} = \boldsymbol{\lambda}dt + \mathbf{\Lambda}d\eta$  [30].

In what follows the resolution of the equations of motion expressed in the form of the equality of measures (27) is briefly presented and discussed in a review manner. The inclusions that are necessary to solve the frictional unilateral contact events in an autonomous multibody system, based on the Newton's impact law combined with the Coulomb's friction law, are also stated. In addition, the force laws are related to the systems' kinematics. The interested reader in the detail description of this formulation is referred to the references [18, 28, 31].

Since the impenetrability condition between colliding bodies is required, let us consider a MBS with  $n$  of frictional unilateral constraints, which can be represented by  $n$  inequalities as,

$$g_{Ni}(\mathbf{q}, t) \geq 0, i = 1, \dots, n \quad (28)$$

where the quantities  $g_{Ni}$  are the normal gap functions of the frictional contacts. They are formulated such that,  $g_{Ni} > 0$  indicates an open or positive contact with an Euclidian distance of the contact points given by the value of  $g_{Ni}$ ,  $g_{Ni} = 0$  corresponds to a closed or active contact, and  $g_{Ni} < 0$  indicates the forbidden overlapping or interpenetration between rigid bodies. A rigorous treatment of the definition of these inequalities, within the framework of multibody systems formulation, is presented and discussed by Pfeiffer and Glocker [32] and Glocker [18].

The set of active contacts in the present work is stated as,

$$H(t) = \{i \mid g_{Ni}(\mathbf{q}, t) = 0\} \quad (29)$$

which singles out the contact(s) at which contact-impact forces may occur.

In order to define the constitutive force laws which relate the contact-impact impulse measures to the system's kinematics  $\mathbf{q}$  and  $\mathbf{u}$ , let us first introduce the normal and tangential relative velocities at the contacts as [33]

$$\gamma_{Ni} = \mathbf{w}_{Ni}^T \mathbf{u} + \tilde{w}_{Ni} \quad (30)$$

$$\gamma_{Ti} = \mathbf{w}_{Ti}^T \mathbf{u} + \tilde{w}_{Ti} \quad (31)$$

where  $\mathbf{w}_{Ni}$  and  $\mathbf{w}_{Ti}$  represent the generalized normal and tangential force directions, respectively, and  $\tilde{w}_{Ni}$  and  $\tilde{w}_{Ti}$  are rheonomic terms [18].

The equations of motion (27) can now be complemented with constitutive laws for normal and tangential contact-impact forces. In the present study, a unilateral version of the Newton's impact law is considered for the normal direction with local coefficient of restitution  $\varepsilon_{Ni} \in [0, 1]$ . The Coulomb's friction law is used for the tangential direction with coefficient of friction  $\mu_i$ , which is complemented by a tangential coefficient of restitution  $\varepsilon_{Ti} \in [0, 1]$ . For the case of a completely elastic contact the coefficient of restitution is equal to unity, while for a perfectly inelastic contact the coefficient of restitution assumes the value of zero.

It is important to note that for the Newton's impact law, the impact, which causes the sudden change in the relative velocity, is accompanied by a normal contact impulse  $d\mathbf{P}_N > 0$ . Suppose that, for any reason, the contact does not participate in the impact, that is, that value of the normal contact impulse is zero, although the contact is closed. This situation happens normally for multiple contact scenarios. Therefore, for this case, we allow the post-impact relative velocity to be higher than the value prescribed by Newton's impact law, with the intent to express that the contact is superfluous and could be removed without changing the contact-impact process. Thus, in order to account for these possibilities, two parameters are defined as [33]

$$\xi_{Ni} := \gamma_{Ni}^+ + \varepsilon_{Ni} \gamma_{Ni}^- \quad (32)$$

$$\xi_{Ti} := \gamma_{Ti}^+ + \varepsilon_{Ti} \gamma_{Ti}^- \quad (33)$$

where  $(\gamma_{Ni}^+, \gamma_{Ti}^-) := (\gamma_{Ni}, \gamma_{Ti}) (\mathbf{u}^\pm)$ .

Thus, the normal and tangential impact laws can be stated as two inclusions

$$-d\mathbf{P}_{Ni} \in \text{Upr}(\xi_{Ni}) \quad (34)$$

$$-d\mathbf{P}_{Ti} \in \mu_i d\mathbf{P}_{Ni} \text{Sgn}(\xi_{Ti}) \quad (35)$$

Finally, the complete description of the dynamics of nonsmooth system, which accounts for both impact and impact-free phases, is given by Eqs. (27)–(35). This problem can be solved by using the Moreau's time-stepping method, which is presented and discussed in the next section.

## 5 Moreau's Time-Stepping Method

The time-stepping methods provide a discrete numerical scheme suitable for the simulation of nonsmooth systems [30–35]. These methods are widely used due to their simplicity to implement and their robustness. The time-stepping schemes are based on a time-discretization of the system dynamics. The whole set of discretized equations and constraints is used to compute the next state of the motion. Among the various time-stepping methods available in the literature, the Moreau's mid-point method is one of the most popular and is considered in the present work [28]. The equality of measures (27) together with the set-valued force laws (34) and (35) form a measure differential inclusion which describes the time-evolution of a multi-body system with discontinuities in the generalized velocities, that is, a nonsmooth dynamical system. A general way to solve this mathematical problem consists of applying the Moreau's time-stepping method, which does not make use of the classical equations of motion, which relate the accelerations to forces, but considers the equations of motion at the velocity level (27). The first step of the Moreau's approach consists of the time-discretization of the measure differential equation. Integrating Eq. (27) over a small finite time interval  $\Delta t$ , of which initial and end points are denoted by the indices  $A$  and  $E$ , yields the following terms

$$\int_{\Delta t} \mathbf{M} d\mathbf{u} \approx \mathbf{M}_M \Delta \mathbf{u} = \mathbf{M}_M (\mathbf{u}_E - \mathbf{u}_A), \quad \mathbf{M}_M = \mathbf{M}(\mathbf{q}_M, t_M) \quad (36)$$

$$\int_{\Delta t} \mathbf{h} dt = \Delta \mathbf{h} \approx \mathbf{h}_M \Delta t, \quad \mathbf{h}_M = \mathbf{h}(\mathbf{q}_M, \mathbf{u}_A, t_M) \quad (37)$$

$$\int_{\Delta t} \mathbf{W}_N d\mathbf{P}_N = \mathbf{W}_{NM} \mathbf{P}_N, \quad \mathbf{W}_{NM} = \mathbf{W}_N(\mathbf{q}_M, t_M) \quad (38)$$

$$\int_{\Delta t} \mathbf{W}_T d\mathbf{P}_T = \mathbf{W}_{TM} \mathbf{P}_T, \quad \mathbf{W}_{TM} = \mathbf{W}_T(\mathbf{q}_M, t_M) \quad (39)$$

in which  $t_M$  is the midpoint time instant of the compact time interval  $[t_A, t_E]$  and  $\mathbf{q}_M = \mathbf{q}_A + \frac{1}{2}\mathbf{u}_A\Delta t$  is the midpoint system's position state. It is clear that the midpoint time instant can be evaluated as

$$t_M = t_A + \frac{1}{2}\Delta t \quad (40)$$

Finally, after the above discretization, the equations of motion expressed at the velocity level can be written as [20]

$$\mathbf{M}_M (\mathbf{u}_E - \mathbf{u}_A) - \mathbf{h}_M \Delta t - \mathbf{W}_{NM} \mathbf{P}_N - \mathbf{W}_{TM} \mathbf{P}_T = \mathbf{0} \quad (41)$$

together with the set-valued contact/impact laws

$$-\mathbf{P}_N \in \text{Upr}(\xi_N) \Leftrightarrow -\mathbf{P}_N \in N_{C_N}(\xi_N) \quad (42)$$

$$-\mathbf{P}_T \in \mu \mathbf{P}_N \text{Sgn}(\xi_T) \Leftrightarrow -\mathbf{P}_T \in N_{C_T(\mathbf{P}_N)}(\xi_T) \quad (43)$$

This set of algebraic inclusions can be solved with a linear complementarity problem (LCP) formulation or by an augmented Lagrangian approach (ALA) [17]. The velocity  $\mathbf{u}_E$ , at the end of time-step  $t_E = t_A + \Delta t$ , is subsequently calculated by using Eq. (41). Finally, the positions at the end of the time step are calculated by

$$\mathbf{q}_E = \mathbf{q}_M + \frac{1}{2}\Delta t \mathbf{u}_E \quad (44)$$

Note that Eq. (42) applies only to active set-valued force laws,  $i \in H(t)$ , i.e. set-valued force laws that can be described at the velocity level. As friction elements are naturally defined at the velocity level, they are always active and can always be described by (43). Considering unilateral contacts, Moreau's midpoint algorithm calculates the contact distances  $g_{Ni}$  of all unilateral contacts at the midpoint  $\mathbf{q}_M$  in order to evaluate whether these are active ( $g_{Ni} \leq 0$ ) or not ( $g_{Ni} > 0$ ). Only active unilateral contacts can be described by inclusion (42). Unilateral contacts that are non-active, thus open, are disregarded because it is assumed that their contact force contribution is equal to zero. Figure 4 shows the flowchart of the general computational strategy, based on the Moreau's time-stepping method, to solve the equations of motion for multibody systems with frictional unilateral constraints.

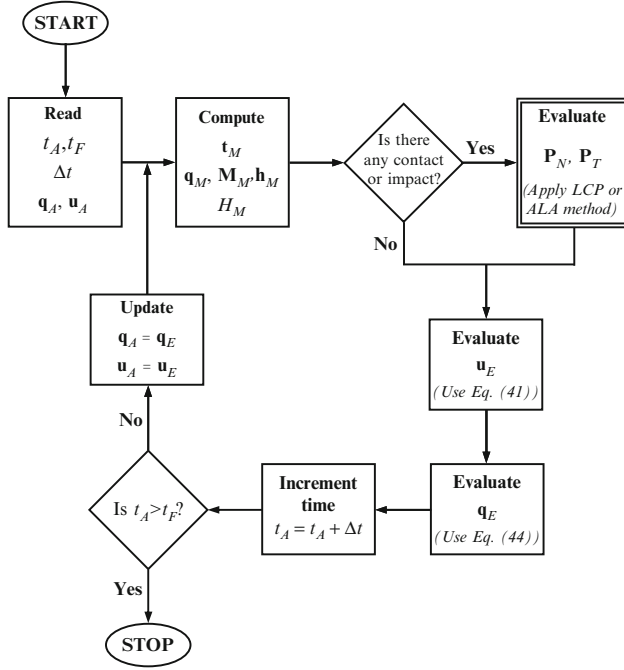
In what follows, the LCP formulation to solve the contact-impact problem of multibody systems with frictional unilateral constraints is presented, which closely follows the work by Glocker and Studer [20]. In order to set up the LCP, let us first introduce the following matrix notation

$$\mathbf{W}_{NM} := \text{mat}(\mathbf{w}_{Ni}(\mathbf{q}_M, t_M)) \in \mathbb{R}^{f,i}, \quad i \in H \quad (45)$$

$$\mathbf{W}_{TM} := \text{mat}(\mathbf{w}_{Ti}(\mathbf{q}_M, t_M)) \in \mathbb{R}^{f,i}, \quad i \in H \quad (46)$$

$$\tilde{\mathbf{w}}_{NM} := \text{col}(\tilde{w}_{Ni}(\mathbf{q}_M, t_M)) \in \mathbb{R}^i, \quad i \in H \quad (47)$$

$$\tilde{\mathbf{w}}_{TM} := \text{col}(\tilde{w}_{Ti}(\mathbf{q}_M, t_M)) \in \mathbb{R}^i, \quad i \in H \quad (48)$$



**Fig. 4** Flowchart of the computational procedure for the solution of the equations of motion of constrained rigid multibody systems with frictional unilateral constraints

$$\mathbf{P}_N := \text{col}(P_{Ni}) \in \mathbb{R}^i, \quad i \in H \quad (49)$$

$$\mathbf{P}_T := \text{col}(P_{Ti}) \in \mathbb{R}^i, \quad i \in H \quad (50)$$

$$\boldsymbol{\gamma}_{NE} := \text{col}(\gamma_{NEi}) \in \mathbb{R}^i, \quad i \in H \quad (51)$$

$$\boldsymbol{\gamma}_{TE} := \text{col}(\gamma_{TEi}) \in \mathbb{R}^i, \quad i \in H \quad (52)$$

$$\boldsymbol{\gamma}_{NA} := \text{col}(\gamma_{NAi}) \in \mathbb{R}^i, \quad i \in H \quad (53)$$

$$\boldsymbol{\gamma}_{TA} := \text{col}(\gamma_{TAi}) \in \mathbb{R}^i, \quad i \in H \quad (54)$$

$$\boldsymbol{\xi}_N := \text{col}(\xi_{Ni}) \in \mathbb{R}^i, \quad i \in H \quad (55)$$

$$\boldsymbol{\xi}_T := \text{col}(\xi_{Ti}) \in \mathbb{R}^i, \quad i \in H \quad (56)$$

$$\boldsymbol{\varepsilon}_N := \text{diag}(\epsilon_{Ni}) \in \mathbb{R}^i, \quad i \in H \quad (57)$$

$$\boldsymbol{\varepsilon}_T := \text{diag}(\epsilon_{Ti}) \in \mathbb{R}^i, \quad i \in H \quad (58)$$

$$\boldsymbol{\mu} := \text{diag}(\mu_i) \in \mathbb{R}^i, \quad i \in H \quad (59)$$

Thus, the contact-impact problem of nonsmooth systems can be summarized by the following mathematical relations

$$\mathbf{M}_M (\mathbf{u}_E - \mathbf{u}_A) - \mathbf{h}_M \Delta t - \mathbf{W}_{NM} \mathbf{P}_N - \mathbf{W}_{TM} \mathbf{P}_T = \mathbf{0} \quad (60)$$

$$\boldsymbol{\gamma}_{NE} = \mathbf{W}_{NM}^T \mathbf{u}_E + \tilde{\mathbf{w}}_{NM} \quad (61)$$

$$\boldsymbol{\gamma}_{TE} = \mathbf{W}_{TM}^T \mathbf{u}_E + \tilde{\mathbf{w}}_{TM} \quad (62)$$

$$\boldsymbol{\gamma}_{NA} = \mathbf{W}_{NM}^T \mathbf{u}_A + \tilde{\mathbf{w}}_{NM} \quad (63)$$

$$\boldsymbol{\gamma}_{TA} = \mathbf{W}_{TM}^T \mathbf{u}_A + \tilde{\mathbf{w}}_{TM} \quad (64)$$

$$\boldsymbol{\xi}_N = \boldsymbol{\gamma}_{NE} + \boldsymbol{\varepsilon}_N \boldsymbol{\gamma}_{NA} \quad (65)$$

$$\boldsymbol{\xi}_T = \boldsymbol{\gamma}_{TE} + \boldsymbol{\varepsilon}_T \boldsymbol{\gamma}_{TA} \quad (66)$$

$$-\mathbf{P}_N \in \text{Upr}(\boldsymbol{\xi}_N) \quad (67)$$

$$-\mathbf{P}_T \in \mu \mathbf{P}_N \text{Sgn}(\boldsymbol{\xi}_T) \quad (68)$$

The values of  $\boldsymbol{\gamma}_{NA}$  and  $\boldsymbol{\gamma}_{TA}$  can be evaluated by using Eqs. (63) and (64), respectively, since the velocities  $\mathbf{u}_A$  are known at the left endpoint of the time interval. Introducing now Eqs. (61) and (52) into Eqs. (65) and (66) yields

$$\boldsymbol{\xi}_N = \mathbf{W}_{NM}^T \mathbf{u}_E + (\tilde{\mathbf{w}}_{NM} + \boldsymbol{\varepsilon}_N \boldsymbol{\gamma}_{NA}) \quad (69)$$

$$\boldsymbol{\xi}_T = \mathbf{W}_{TM}^T \mathbf{u}_E + (\tilde{\mathbf{w}}_{TM} + \boldsymbol{\varepsilon}_T \boldsymbol{\gamma}_{TA}) \quad (70)$$

Now, it should be mentioned that the inclusions for the contact-impact force laws need to be formulated as complementarity conditions. Thus, the unilateral primitive of Eq. (67) results in

$$-\mathbf{P}_N \in \text{Upr}(\boldsymbol{\xi}_N) \Leftrightarrow \mathbf{P}_N \geq \mathbf{0}, \boldsymbol{\xi}_N \geq \mathbf{0}, \mathbf{P}_N^T \boldsymbol{\xi}_N = 0 \quad (71)$$

In turn, the relay function (68) have to be decomposed into two Upr's to achieve the desired complementarity conditions. Thus, Eq. (68) yields

$$\begin{aligned} & -\mathbf{P}_T \in \mu \mathbf{P}_N \text{Sgn}(\boldsymbol{\xi}_T) \\ \Leftrightarrow \exists \boldsymbol{\xi}_R, \boldsymbol{\xi}_L \text{ s.t. } & \begin{cases} \mu \mathbf{P}_N + \mathbf{P}_T \geq \mathbf{0}, & \boldsymbol{\xi}_R \geq \mathbf{0}, & (\mu \mathbf{P}_N + \mathbf{P}_T)^T \boldsymbol{\xi}_R = 0 \\ \mu \mathbf{P}_N - \mathbf{P}_T \geq \mathbf{0}, & \boldsymbol{\xi}_L \geq \mathbf{0}, & (\mu \mathbf{P}_N - \mathbf{P}_T)^T \boldsymbol{\xi}_L = 0 \\ & \boldsymbol{\xi}_T = \boldsymbol{\xi}_R - \boldsymbol{\xi}_L \end{cases} \end{aligned} \quad (72)$$

in which the step height is  $[-\mu \mathbf{P}_N, +\mu \mathbf{P}_N]$ . In addition, to abbreviate the complementarity conditions of Eq. (72) the impulsive friction saturations  $\mathbf{P}_R$  and  $\mathbf{P}_L$  are defined as [18]

$$\mathbf{P}_R := \mu \mathbf{P}_N + \mathbf{P}_T, \quad \mathbf{P}_R \in \mathbb{R}^i \quad (73)$$

$$\mathbf{P}_L := \mu \mathbf{P}_N - \mathbf{P}_T, \quad \mathbf{P}_L \in \mathbb{R}^i \quad (74)$$

together with

$$\boldsymbol{\xi}_T = \boldsymbol{\xi}_R - \boldsymbol{\xi}_L, \boldsymbol{\xi}_R, \boldsymbol{\xi}_L \in \mathbb{R}^i \quad (75)$$

The whole set of complementarity conditions of Eq. (72) can be rewritten as

$$\mathbf{0} \leq \begin{pmatrix} \xi_N \\ \xi_R \\ \mathbf{P}_L \end{pmatrix} \perp \begin{pmatrix} \mathbf{P}_N \\ \mathbf{P}_R \\ \xi_L \end{pmatrix} \geq \mathbf{0} \quad (76)$$

The reason for the special arrangement of  $\mathbf{P}_L$  and  $\xi_L$  in Eq. (76), must be sought in optimization theory. Without this special arrangement, one is not able to be set up the LCP formulation without additional matrix inversion processes [18]. Since the variables  $\xi_T$ ,  $\mathbf{P}_T$  and  $\mathbf{u}_E$  are not included in (28), they have to be eliminated. Thus, combining Eqs. (60) and (73), yields

$$\mathbf{M}_M (\mathbf{u}_E - \mathbf{u}_A) - \mathbf{h}_M \Delta t - (\mathbf{W}_{NM} - \mathbf{W}_{TM}\boldsymbol{\mu}) \mathbf{P}_N - \mathbf{W}_{TM}\mathbf{P}_R = \mathbf{0} \quad (77)$$

Substituting now Eq. (75) into Eq. (70) results in

$$\xi_R = \mathbf{W}_{TM}^T \mathbf{u}_E + (\tilde{\mathbf{w}}_{TM} + \boldsymbol{\varepsilon}_T \boldsymbol{\gamma}_{TA}) + \xi_L \quad (78)$$

The elimination of variable  $\mathbf{P}_T$  can be done through the combination of Eqs. (73) and (74), which can be written as

$$\mathbf{P}_L = 2\boldsymbol{\mu} P_N - \mathbf{P}_R \quad (79)$$

Since the inversion of mass matrix  $\mathbf{M}$  is always possible, Eq. (77) can be solved for  $\mathbf{u}_E$

$$\mathbf{u}_E = \mathbf{u}_A + \mathbf{M}_M^{-1} \mathbf{h}_M \Delta t + \mathbf{M}_M^{-1} (\mathbf{W}_{NM} - \mathbf{W}_{TM}\boldsymbol{\mu}) \mathbf{P}_N + \mathbf{M}_M^{-1} \mathbf{W}_{TM} \mathbf{P}_R \quad (80)$$

Now, Eqs. (63) and (64) are used to express  $\mathbf{W}_{NM}^T \mathbf{u}_A$  and  $\mathbf{W}_{TM}^T \mathbf{u}_A$  in terms of  $\boldsymbol{\gamma}_{NA}$  and  $\boldsymbol{\gamma}_{TA}$

$$\mathbf{W}_{NM}^T \mathbf{u}_A = \boldsymbol{\gamma}_{NA} - \tilde{\mathbf{w}}_{NM} \quad (81)$$

$$\mathbf{W}_{TM}^T \mathbf{u}_A = \boldsymbol{\gamma}_{TA} - \tilde{\mathbf{w}}_{TM} \quad (82)$$

Introducing Eqs. (80)–(82) into Eqs. (69) and (78), yields

$$\begin{aligned} \xi_N &= \mathbf{W}_{NM}^T \mathbf{M}_M^{-1} \mathbf{h}_M \Delta t + \mathbf{W}_{NM}^T \mathbf{M}_M^{-1} (\mathbf{W}_{NM} - \mathbf{W}_{TM}\boldsymbol{\mu}) \mathbf{P}_N \\ &\quad + \mathbf{W}_{NM}^T \mathbf{M}_M^{-1} \mathbf{W}_{TM} \mathbf{P}_R + (\mathbf{I} + \boldsymbol{\varepsilon}_N) \boldsymbol{\gamma}_{NA} \end{aligned} \quad (83)$$

$$\begin{aligned} \xi_R &= \mathbf{W}_{TM}^T \mathbf{M}_M^{-1} \mathbf{h}_M \Delta t + \mathbf{W}_{TM}^T \mathbf{M}_M^{-1} (\mathbf{W}_{NM} - \mathbf{W}_{TM}\boldsymbol{\mu}) \mathbf{P}_N \\ &\quad + \mathbf{W}_{TM}^T \mathbf{M}_M^{-1} \mathbf{W}_{TM} \mathbf{P}_R + (\mathbf{I} + \boldsymbol{\varepsilon}_T) \boldsymbol{\gamma}_{TA} + \xi_L \end{aligned} \quad (84)$$

Thus, Eqs. (83), (84) and (79) can be written in a matrix form as

$$\begin{pmatrix} \xi_N \\ \xi_R \\ \mathbf{P}_L \end{pmatrix} = \begin{pmatrix} \mathbf{W}_{NM}^T \mathbf{M}_M^{-1} (\mathbf{W}_{NM} - \mathbf{W}_{TM} \boldsymbol{\mu}) & \mathbf{W}_{NM}^T \mathbf{M}_M^{-1} \mathbf{W}_{TM} & \mathbf{0} \\ \mathbf{W}_{TM}^T \mathbf{M}_M^{-1} (\mathbf{W}_{NM} - \mathbf{W}_{TM} \boldsymbol{\mu}) & \mathbf{W}_{TM}^T \mathbf{M}_M^{-1} \mathbf{W}_{TM} & \mathbf{I} \\ 2\boldsymbol{\mu} & -\mathbf{I} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{P}_N \\ \mathbf{P}_R \\ \xi_L \end{pmatrix} \\
+ \begin{pmatrix} \mathbf{W}_{NM}^T \mathbf{M}_M^{-1} \mathbf{h}_M \Delta t + (\mathbf{I} + \boldsymbol{\varepsilon}_N) \boldsymbol{\gamma}_{NA} \\ \mathbf{W}_{TM}^T \mathbf{M}_M^{-1} \mathbf{h}_M \Delta t + (\mathbf{I} + \boldsymbol{\varepsilon}_T) \boldsymbol{\gamma}_{TA} \\ \mathbf{0} \end{pmatrix} \quad (85)$$

Equations (85) together with the conditions (76) form the LCP for the contact-impact analysis of multibody systems with frictional unilateral constraints. The LCP (85) is solved in each integration time step. Then, the velocities  $\mathbf{u}_E$  and positions  $\mathbf{q}_E$  for the subsequent time steps are obtained from Eqs. (80) and (44).

Figure 5 illustrates the Moreau's time-stepping method with an LCP formulation developed under the framework of MBS formulation.

## 4 Demonstrative Application to a Slider-Crank Mechanism

This section deals with the dynamic modeling and analysis of a planar slider-crank mechanism with a translation clearance joint. This multibody mechanical system consists of four rigid bodies, which represent the ground, the crank, the connecting rod and the slider. The body numbers and their center of mass are shown in Fig. 6. The ground, the crank, the connecting rod and the slider are constrained via ideal revolute joints. The center of mass of each body is considered to be located at the mid distance of the bodies' total length. The translational clearance joint is composed by a guide and a slider. This joint has a finite clearance, which is constant along the length of the slider.

Figure 7 shows a translational clearance joint. The clearance  $c$  is defined as the difference between the distance of the guide and the slider surfaces. The geometric characteristics of the translational clearance joint are the slider length  $2a$ , the slider width  $2b$ , and the distance between the guide surfaces  $d$ . In an ideal translational joint the two bodies translate with respect to each other parallel to the line of translation, so that, there is neither rotation between the bodies nor a relative translation motion in the direction perpendicular to the axis of the joint. The existence of a clearance in a translational joint introduces two extra degrees of freedom. Hence, the slider can move 'freely' inside the guide limits, until it reaches the guide surfaces.

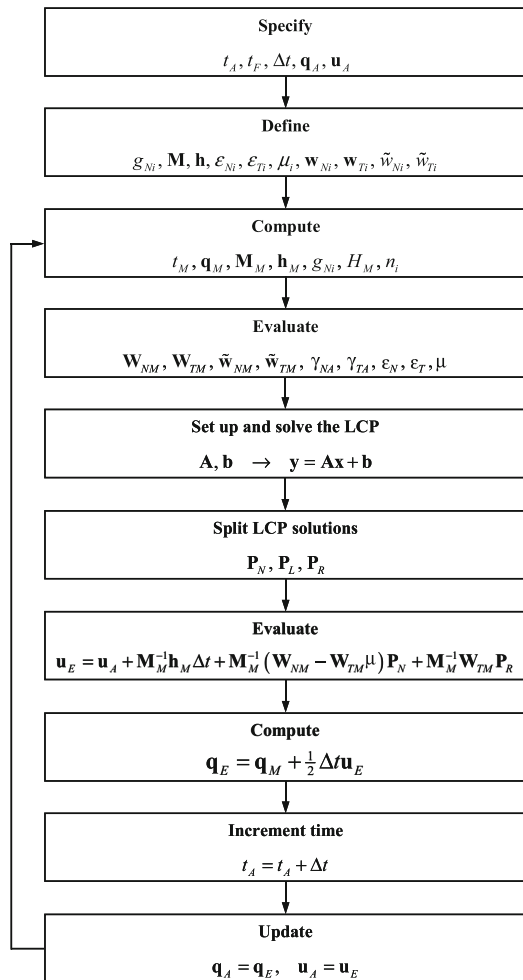
The modeling of translational clearance joints is a complex task, due to the several possible contact configurations between the slider and guide. Figure 8 illustrates four different scenarios for the slider configuration relative to guide surface, namely: (i) No contact between the two elements: the slider is in free flight motion inside the guide; (ii) one corner of the slider is in contact with the guide surface; (iii) two adjacent slider corners are in contact with the guide surface, which implies that a face of slider is in contact with the guide surface; (iv) two opposite slider corners are in



contact with the guide surface. The conditions for switching from one case to another depend on the system's dynamic response as well as on the material colliding properties.

In order for the translational clearance joint to be simulated in the multibody system environment, is it first required that the system's equations of motion be derived. In this work the Lagrange's equation of second type is used and it can be written as [36]

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\mathbf{q}}_i} \right) - \frac{\partial L}{\partial \mathbf{q}_i} = 0, i = 1, \dots, f \quad (86)$$



**Fig. 5** Flowchart of the Moreau's time-stepping algorithm with an LCP formulation

where  $L$  is the Lagrangian of the system, that is, the difference between kinetic and potential energies, expressed in terms of the generalized coordinates and their time derivatives.

Since the slider-crank mechanism represented in Fig. 6 has three degrees of freedom, three is also the number of generalized coordinates that uniquely represent the system's configuration. Furthermore, the crank, the connecting rod and the slider have masses  $m_i$  and moments of inertia with respect to the principal central axes perpendicular to the plane of motion  $J_i$ , where  $i = 1, 2$  and 3. Thus, the vector of generalized coordinates and velocities are defined as

$$\mathbf{q} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} \quad (87)$$

$$\mathbf{u} = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}, \quad \text{with } \dot{\mathbf{q}} = \mathbf{u} \text{ a.e.} \quad (88)$$

Thus, applying the Lagrange's equation to slider-crank mechanism yields [37]

$$\begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix} \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{pmatrix} = \begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix} \quad (89)$$

in which

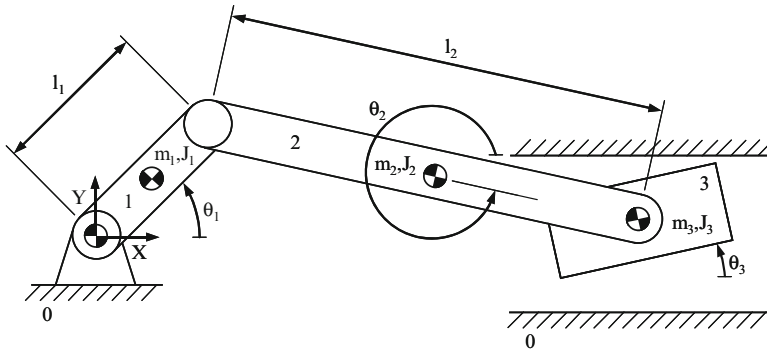
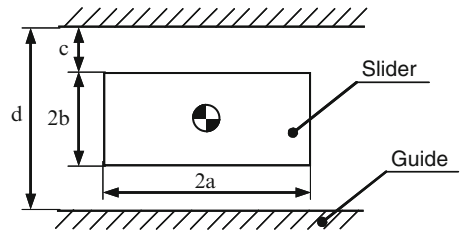
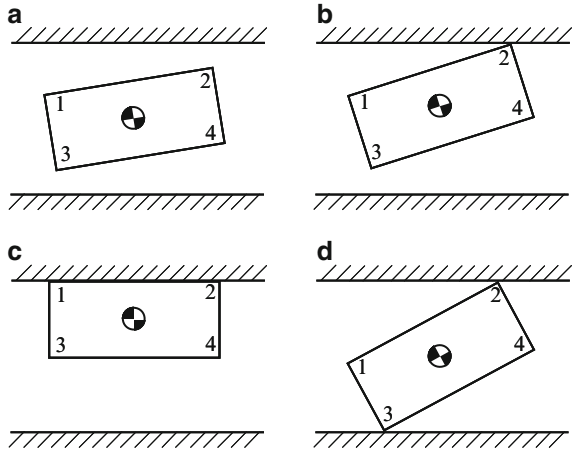


Fig. 6 Slider-crank mechanism with a translational clearance joint

**Fig. 7** Translational joint with clearance that is, the slider and guide



**Fig. 8** Different scenarios for the slider and guide interaction: (a) no contact; (b) one corner in contact with the guide; (c) two adjacent corners in contact with guide; (d) two opposite corners in contact with guide



$$M_{11} = J_1 + \left( \frac{1}{4}m_1 + m_2 + m_3 \right) l_1^2 \quad (90)$$

$$M_{12} = M_{21} = \left( \frac{1}{2}m_2 + m_3 \right) l_1 l_2 \cos(\theta_2 - \theta_1) \quad (91)$$

$$M_{13} = M_{31} = M_{23} = M_{32} = 0 \quad (92)$$

$$M_{22} = J_2 + \left( \frac{1}{4}m_2 + m_3 \right) l_2^2 \quad (93)$$

$$M_{33} = J_3 \quad (94)$$

$$h_1 = \left( \frac{1}{2}m_2 + m_3 \right) l_1 l_2 \sin(\theta_2 - \theta_1) \dot{\theta}_2^2 - \left( \frac{1}{2}m_1 + m_2 + m_3 \right) g l_1 \cos \theta_1 \quad (95)$$

$$h_2 = - \left( \frac{1}{2}m_2 + m_3 \right) l_1 l_2 \sin(\theta_2 - \theta_1) \dot{\theta}_1^2 - \left( \frac{1}{2}m_2 + m_3 \right) g l_2 \cos \theta_2 \quad (96)$$

$$h_3 = 0 \quad (97)$$

In order to determine the gap functions let us consider Fig. 9 where a generic position of the slider inside the guide is illustrated with the purpose to represent the closed kinematic chain of each potential contact point.

From analysis of Fig. 9 and considering the system kinematics, the mathematical expressions of the gap functions can be written as [37]

$$g_{N1} = \frac{d}{2} - l_1 \sin \theta_1 - l_2 \sin \theta_2 + a \sin \theta_3 - b \cos \theta_3 \quad (98)$$

$$g_{T1} = l_1 \cos \theta_1 + l_2 \cos \theta_2 - a \cos \theta_3 - b \sin \theta_3 \quad (99)$$

$$g_{N2} = \frac{d}{2} - l_1 \sin \theta_1 - l_2 \sin \theta_2 - a \sin \theta_3 - b \cos \theta_3 \quad (100)$$

$$g_{T2} = l_1 \cos \theta_1 + l_2 \cos \theta_2 + a \cos \theta_3 - b \sin \theta_3 \quad (101)$$

$$g_{N3} = \frac{d}{2} + l_1 \sin \theta_1 + l_2 \sin \theta_2 - a \sin \theta_3 - b \cos \theta_3 \quad (102)$$

$$g_{T3} = l_1 \cos \theta_1 + l_2 \cos \theta_2 - a \cos \theta_3 + b \sin \theta_3 \quad (103)$$

$$g_{N4} = \frac{d}{2} + l_1 \sin \theta_1 + l_2 \sin \theta_2 + a \sin \theta_3 - b \cos \theta_3 \quad (104)$$

$$g_{T4} = l_1 \cos \theta_1 + l_2 \cos \theta_2 + a \cos \theta_3 + b \sin \theta_3 \quad (105)$$

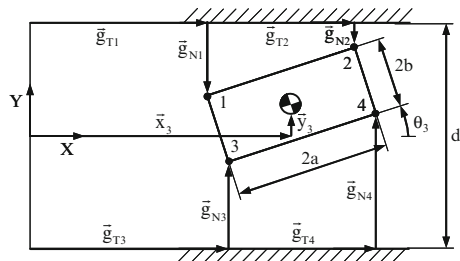
Then, the  $\mathbf{w}$  vectors and of the  $\tilde{w}$  scalars associated with each contact point can be obtained as

$$\mathbf{w}_{N1} = \frac{\partial g_{N1}}{\partial \mathbf{q}} = \begin{pmatrix} -l_1 \cos \theta_1 \\ -l_2 \cos \theta_2 \\ a \cos \theta_3 + b \sin \theta_3 \end{pmatrix} \quad (106)$$

$$\mathbf{w}_{T1} = \frac{\partial g_{T1}}{\partial \mathbf{q}} = \begin{pmatrix} -l_1 \sin \theta_1 \\ -l_2 \sin \theta_2 \\ a \sin \theta_3 - b \cos \theta_3 \end{pmatrix} \quad (107)$$

$$\mathbf{w}_{N2} = \frac{\partial g_{N2}}{\partial \mathbf{q}} = \begin{pmatrix} -l_1 \cos \theta_1 \\ -l_2 \cos \theta_2 \\ -a \cos \theta_3 + b \sin \theta_3 \end{pmatrix} \quad (108)$$

**Fig. 9** Generic position of the slider inside the guide where the distance between guide upper and lower surfaces is exaggerated for illustration purpose



$$\mathbf{w}_{T2} = \frac{\partial g_{T2}}{\partial \mathbf{q}} = \begin{pmatrix} -l_1 \sin \theta_1 \\ -l_2 \sin \theta_2 \\ -a \sin \theta_3 - b \cos \theta_3 \end{pmatrix} \quad (109)$$

$$\mathbf{w}_{N3} = \frac{\partial g_{N3}}{\partial \mathbf{q}} = \begin{pmatrix} l_1 \cos \theta_1 \\ l_2 \cos \theta_2 \\ -a \cos \theta_3 + b \sin \theta_3 \end{pmatrix} \quad (110)$$

$$\mathbf{w}_{T3} = \frac{\partial g_{T3}}{\partial \mathbf{q}} = \begin{pmatrix} -l_1 \sin \theta_1 \\ -l_2 \sin \theta_2 \\ a \sin \theta_3 + b \cos \theta_3 \end{pmatrix} \quad (111)$$

$$\mathbf{w}_{N4} = \frac{\partial g_{N4}}{\partial \mathbf{q}} = \begin{pmatrix} l_1 \cos \theta_1 \\ l_2 \cos \theta_2 \\ a \cos \theta_3 + b \sin \theta_3 \end{pmatrix} \quad (112)$$

$$\mathbf{w}_{T4} = \frac{\partial g_{T4}}{\partial \mathbf{q}} = \begin{pmatrix} -l_1 \sin \theta_1 \\ -l_2 \sin \theta_2 \\ -a \sin \theta_3 + b \cos \theta_3 \end{pmatrix} \quad (113)$$

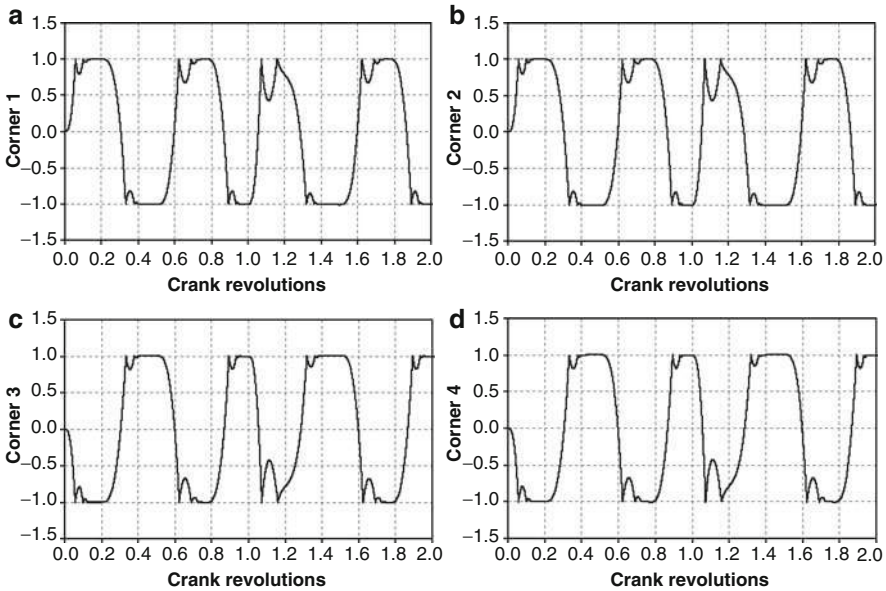
$$\tilde{w}_{N1} = \tilde{w}_{T1} = \tilde{w}_{N2} = \tilde{w}_{T2} = \tilde{w}_{N3} = \tilde{w}_{T3} = \tilde{w}_{N4} = \tilde{w}_{T4} = 0 \quad (114)$$

The geometrical characteristics, the inertial properties, the force elements, the contact parameters and the initial conditions necessary to perform the dynamic analysis of the slider-crank mechanism with a translational clearance joint are listed in Table 1.

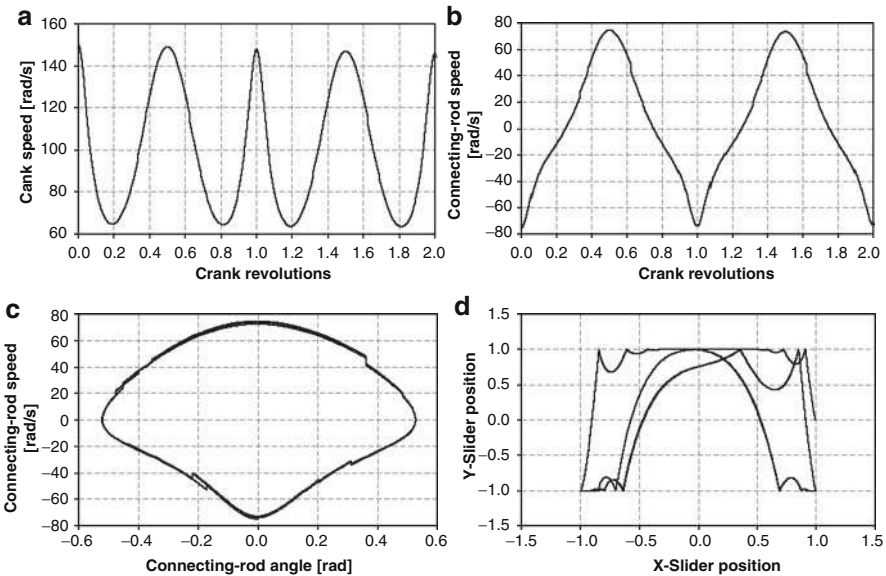
Figure 10 shows the corners motion in a dimensionless form for two full crank rotations, in which the free slider motion and contact-impact events can be observed. Figure 11 illustrates the crank speed, the connecting-rod speed and the portraits relative to connecting-rod and slider for two complete crank rotations.

The dimensionless slider trajectories are shown in Fig. 10, where the different types of motion between the slider and guide observed are associated with the different guide-slider configurations, i.e., no contact, impact followed by rebound and permanent contact between the joint elements. The effects of impact between the slider and guide surfaces are also quite visible in the plots of Fig. 11b and c, namely, one can observe the discontinuities in the connecting-rod speed. On the other hand, the smooth changes in the speed indicate that the slider and guide surfaces are in permanent contact for long periods, as it is illustrated in the slider portrait of Fig. 11d.

It should be highlighted that some numerical difficulties can arise when the clearance size is very small, which will lead to the well known drift problem. In these situations, one possible way to overcome those difficulties consists of a projection technique, in which the excessive penetration between the slider and guide surfaces is eliminated in each time step in order to avoid the further interpretation of the bodies. When this scheme is implemented, special attention should be paid to the



**Fig. 10** Dimensionless motion of the slider corners. (a) Corner 1; (b) corner 2; (c) corner 3; (d) corner 4



**Fig. 11** (a) Crank speed; (b) connecting-rod speed; (c) connecting-rod portrait; (d) slider portrait

**Table 1** Parameters used in the dynamic simulation of the slider-crank mechanism

Geometrical characteristics	$l_1 = 0.1530 \text{ m}$ $l_2 = 0.3060 \text{ m}$ $a = 0.0500 \text{ m}$ $b = 0.0250 \text{ m}$ $c = 0.0010 \text{ m}$
Inertial properties	$m_1 = m_2 = 0.0380 \text{ kg}$ $m_3 = 0.0760 \text{ kg}$ $J_1 = 7.4 \times 10^{-5} \text{ kg m}^2$ $J_2 = 5.9 \times 10^{-4} \text{ kg m}^2$ $J_3 = 2.7 \times 10^{-6} \text{ kg m}^2$
Force elements	$g = 9.81 \text{ m/s}^2$
Contact parameters	$\varepsilon_{N1} = \varepsilon_{N2} = \varepsilon_{N3} = \varepsilon_{N4} = 0.4$ $\varepsilon_{T1} = \varepsilon_{T2} = \varepsilon_{T3} = \varepsilon_{T4} = 0.0$ $\mu_1 = \mu_2 = \mu_3 = \mu_4 = 0.01$
Initial conditions	$\theta_{10} = \theta_{20} = \theta_{30} = 0.0 \text{ rad}$ $\omega_{10} = 150.0 \text{ rad/s}$ $\omega_{20} = -75.0 \text{ rad/s}$ $\omega_{30} = 0.0 \text{ rad/s}$

conservation of the systems energy, since it can lead to overestimated total system energy associated with the contact-impact phenomena.

## 5 Conclusions

A comprehensive investigation of contact-impact analysis in multibody systems based on the nonsmooth dynamics approach was presented in this work. The methodology was based on the nonsmooth dynamics approach, in which the interaction of the colliding bodies is modeled with multiple frictional unilateral constraints. The dynamics of rigid multibody systems were stated as an equality of measures, which were formulated at the velocity-impulse level. The equations of motion were complemented with constitutive laws for the forces and impulses in normal and tangential directions. The formulation of the generalized contact-impact kinematics in the normal and tangential directions was performed by obtaining a geometric relation for the gaps of the candidate contact points. The gaps were expressed as functions of the generalized coordinates. The candidate contact points were modeled as hard contacts, being the normal and tangential contact laws formulated as set-valued force laws for frictional unilateral constraints.

In this work, the unilateral constraints were described by a set-valued force law of the type of Signorini's condition, while the frictional contacts were characterized by a set-valued force law of the type of Coulomb's law for dry friction. The result-

ing contact-impact problem was formulated and solved as a linear complementarity problem and with the augmented Lagrangian approach, which were embedded in the Moreau's time-stepping method. Finally, the effectiveness of the presented methodologies was demonstrated through the study of the slider crank mechanism with a translational clearance joint. The main results obtained from this research work showed that the effect of the contact-impact phenomena can have a predictable nonlinear behavior. This nonlinearity aspect is more evident when the system includes friction phenomena. With the knowledge of nonlinearities in multibody systems, chaotic behavior may be eliminated with suitable design and/or parameter changes of a mechanical system. This feature plays a crucial role in the dynamics, design and control of general multibody systems of common application.

**Acknowledgement** This work is supported by the Portuguese Foundation for the Science and Technology under a postdoctoral scholarship (SFRH/BPD/40067/2007).

## References

1. Flores P, Ambrósio J (2004) Revolute joints with clearance in multibody systems. *Comput Struct* 82:1359–1369
2. Flores P, Ambrósio J, Claro JCP, Lankarani HM (2008) Kinematics and dynamics of multibody systems with imperfect joints: models and case studies. *Lecture notes in applied and computational mechanics*, Vol. 34. Springer, Berlin, Heidelberg
3. Liu C-S, Zhang K, Yang R (2007) The FEM analysis and approximate model for cylindrical joints with clearances. *Mech Mach Theory* 42:183–197
4. Crowther AR, Singha R, Zhangb N, Chapman C (2007) Impulsive response of an automatic transmission system with multiple clearances: formulation, simulation and experiment. *J Sound Vibration* 306:444–466
5. Flores P, Ambrósio J, Claro JCP, Lankarani HM, (2008) Translational joints with clearance in rigid multibody systems. *J Comput Nonlinear Dyn* 3(1):0110071–10
6. Awrejcewicz J, Kudra G, Lamarque C-H (2003) Dynamics investigation of three coupled rods with horizontal barrier. *Meccanica* 38:687–698
7. Stoenescu ED, Marghitu DB (2003) Dynamic analysis of a planar rigid-link mechanism with rotating slider joint and clearance. *J Sound Vibration* 266:394–404
8. Farahanchi F, Shaw SW (1994) Chaotic and periodic dynamics of a slider crank mechanism with slider clearance. *J Sound Vibration* 177(3):307–324
9. Thümmel T, Funk K (1991) Multibody modelling of linkage mechanisms including friction, clearance and impact. In: *Proceedings of tenth world congress on the theory of machine and mechanisms*, vol 4, Oulu University, Finland, pp 1375–1386
10. Wilson R, Fawcett JN (1994) Dynamics of slider-crank mechanism with clearance in the sliding bearing. *Mech Mach Theory* 9:61–80
11. Cottle RW, Dantzig GB (1968) Complementary pivot theory of mathematical programming. *Linear Algebra Appl* 1:103–125
12. Cottle RW, Pang JS, Stone RE (1992) *The linear complementarity problem*. Academic, London
13. Rohn J (1993) A note on solvability of a class of linear complementarity problems. *Math Programm* 60(1–3):229–231
14. Leine TI, Brogliato B, Nijmeijer H (2002) Periodic motion and bifurcations induced by the Painlevé paradox. *Eur J Mech A/Solids* 21(5):869–896
15. Pang J, Trinkle JC (1996) Complementarity formulations and existence of solutions of dynamic multi-rigid-body contact problems with Coulomb friction. *Math Programm* 73(2):199–226



16. Stewart DE, Trinkle JC (1996) An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and Coulomb friction. *Int J Numerical Methods Eng* 39(15):2673–2691
17. Leine RI, Nijmeijer H (2004) Dynamics and bifurcations of non-smooth mechanical systems. Lecture notes in applied and computational mechanics, Vol 18. Springer-Verlag, Berlin, Heidelberg, New York
18. Glocker C (2001) Set-valued force laws: dynamics of non-smooth systems. Lecture notes in applied mechanics 1. Springer, Berlin
19. Leine RI, van de Wouw N (2008) Stability and convergence of mechanical systems with unilateral constraints. Lecture notes in applied and computational mechanics, vol 36. Springer, Berlin Heidelberg, New York
20. Glocker C, Studer C (2005) Formulation and preparation for numerical evaluation of linear complementarity systems in dynamics. *Multibody Syst Dyn* 13(4):447–463
21. Signorini A (1933) Sopra alcune questioni di elastostatica. *Atti della Societa Italian per il Progresso della Scienza*
22. Leine RI, van de Wouw N (2008) Stability properties of equilibrium sets of non-linear mechanical systems with dry friction and impact. *Nonlinear Dyn* 51(4):551–583
23. Leine RI, Glocker C (2003) A set-valued force law for spatial Coulomb–Contensou friction. *Eur J Mech – A/Solids* 22(2):193–216
24. Jean M (1999) The non-smooth contact dynamics method. *Comput Methods Appl Mech Eng* 177(3–4):235–257
25. Pfeiffer F, Foerg M, Ulbrich H (2006) Numerical aspects of non-smooth multibody dynamics. *Computer Methods Appl Mech Eng* 195:6891–6908
26. Haug EJ (1989) Computer-aided kinematics and dynamics of mechanical systems, vol I: Basic methods. Allyn & Bacon, Boston, MA
27. Nikravesh P (1988) Computer-aided analysis of mechanical systems. Prentice-Hall, Englewood Cliffs, NJ
28. Moreau JJ (1988) Unilateral contact and dry friction in finite freedom dynamics. In: Moreau JJ, Panagiotopoulos PD (eds) *Non-smooth mechanics and applications CISM courses and lectures*, vol 302. Springer Verlag, Wien, pp 1–82
29. Panagiotopoulos PD, Glocker C (2000) Inequality constraints with elastic impacts in deformable bodies. The convex case. *Arch Appl Mech* 70(5):349–365
30. Förg M, Pfeiffer F, Ulbrich H (2005) Simulation of unilateral constrained systems with many bodies. *Multibody Syst Dyn* 14(2):137–154
31. Moreau JJ (1988) Bounded variation in time, in topics in nonsmooth mechanics. In: Moreau JJ, Panagiotopoulos PD, Strang G (eds) *Topics in nonsmooth mechanics*. Birkhäuser Verlag, Basel, pp 1–74
32. Pfeiffer F, Glocker C (1996) *Multibody dynamics with unilateral contacts*. Wiley, New York
33. Glocker C (2001) On frictionless impact models in rigid-body systems. *Phil Trans: Math Phys Eng Sci* 359:2385–2404
34. Anitescu M, Potra FA, Stewart DE (1999) Time-stepping for three-dimensional rigid body dynamics. *Comput Methods Appl Mech Eng* 177(3):183–197
35. Studer C, Leine RI, Glocker C (2008) Step size adjustment and extrapolation for time-stepping schemes in non-smooth dynamics. *Int J Numerical Methods Eng* 76(11):1747–1781
36. Greenwood DT (1965) *Principles of dynamics*. Prentice-Hall, Inc., Englewood Cliffs, NJ
37. Flores P (2009) Contact-impact analysis in multibody systems based on the nonsmooth dynamics approach. Post Doctoral Report, ETH-Zurich, Switzerland

# Application of General Multibody Methods to Robotics

Janusz Frączek and Marek Wojtyra

**Abstract** In this chapter robotic applications of general multibody system (MBS) simulation methods, based on absolute coordinates formalism, are presented. Three typical problems, often encountered in robotics, are discussed: kinematic analysis with singular configuration detection, simulation of parallel robot dynamics investigated jointly with the robot control systems properties, and finally, simulation of a robot with flexibility effects taken into account. In case of singular configuration detection simplest types of singular configurations are analyzed – turning point and bifurcation point. The second case of MBS application is an example of parallel robot dynamic analysis when model based control is taken into account. The last part of the chapter is devoted to the analysis of complex, flexible power transmission mechanism carried out with general MBS formalism.

## 1 Introduction

Many modern robots are constructed in such a way that they have relatively simple kinematic structure and, as a result, solutions to direct and inverse kinematics problems, formulated usually in joint coordinates, are known in closed (symbolic) form. Moreover, differential equations of robot motion may also be formulated in a problem specific way. Most often algorithms of kinematic or dynamic calculations, which are used by robot controller, are tailored to fit the particular robot. Computer code written especially for specific mechanism structure is simpler and more efficient than a general program. Thus, usually there is no need for general multibody methods application in robot everyday operation.

General multibody methods can be very useful, however, during the robot design process. It is convenient to use these methods, especially in case of analysis or

---

J. Frączek (✉) and M. Wojtyra  
Warsaw University of Technology, Institute of Aeronautics and Applied Mechanics,  
Nowowiejska 24, 00-665 Warsaw, Poland  
e-mail: [jfraczek@meil.pw.edu.pl](mailto:jfraczek@meil.pw.edu.pl); [mwojtyra@meil.pw.edu.pl](mailto:mwojtyra@meil.pw.edu.pl)

synthesis of robot with complicated structure (with many closed kinematic loops), since they do not require closed form kinematics solution. Moreover, kinematic and dynamic equations are formulated automatically and no tedious derivation of formulas is required. The general multibody methods can be used, e.g., for kinematic structure synthesis and analysis, for virtual tests of control system, as well as for calculating loads and stresses in robot elements.

It is worth noting that quite often general and commonly used multibody methods are based on absolute coordinates formalism. This formalism is not popular in robotics, since joint coordinates are most commonly used in that field.

In this chapter robotic applications of general multibody simulation methods, based on absolute coordinates formalism (which is commonly used in practical calculations and in popular software packages) are presented. Three typical problems, often encountered in robotics, are discussed: kinematic analysis with singular configurations detection, simulation of a parallel robot dynamics investigated jointly with the robot control systems properties, and finally, simulation of a robot with flexibility effects taken into account.

## **2 Absolute Coordinates Approach to Robot Kinematic Analysis and Singular Configuration Detection**

Singularities of manipulators are a widely recognized issue in robotics. Robot singularities usually mean kinematic singularities where the number of robot degrees of freedom locally changes. The singularities were intensively studied in robotics, e.g. [1–3], but the classification and global description of singular configurations exists only for special cases. Vast majority of mentioned algorithms is devoted to singularity analysis of robots and manipulators with known, relatively simple structure and with small number of kinematic loops. Many authors investigated special problems of singularities in case of analysis of parallel manipulators. Singularities of parallel manipulators were classified in terms of kinematic input-output relations and this classification is commonly applied [4].

The singular configuration detection is also an important topic in general multibody simulations. Unfortunately, the ideas and methods of singularity detection and classification developed in robotics cannot be directly transferred to topology-independent kinematic analysis of general, complicated multibody systems analyzed in absolute coordinates. It should be emphasized that number of papers devoted to singularity detection and analysis in case of kinematic simulation of general multibody system is limited.

In this section we will show an attempt of novel application of numerical continuation algorithm [5] to kinematic analysis of a robotic system. This algorithm uses local coordinate parameterization instead of time parameterization which is commonly used in multibody system simulation. The algorithm is suitable for kinematic analysis of a general multibody system described in absolute coordinates. The simplest cases of singularities may be detected using mentioned ideas.

We will illustrate developed methods on the example of preliminary kinematic studies of a complicated robotic construction [6] in both singularity-free and singular configurations.

## 2.1 Theoretical Background

The algorithm of kinematic analysis in absolute coordinates is usually based on the trajectory tracing using time as the independent parameter and the classical Newton iteration scheme [7]. In case of analysis of complicated mechanism, simulation usually fails in singular positions and the reason of that cannot be easily detected by the user. This creates needs for singularities detection and analysis. Moreover, singular positions analysis can provide interesting information, relevant for the mechanism structural synthesis and control system synthesis.

The collection of all constraints induced by the joints present in MBS is denoted by:

$$\Phi^k(\mathbf{q}, t) = \mathbf{0}, \quad (1)$$

where  $\mathbf{q}$  is the vector of absolute generalised coordinates [7].

The number of scalar equations in (1) is equal to  $l$  and the number of generalized coordinates is equal to  $N = 6m$ , where  $m$  is the number of rigid bodies (three variables for rotation parameterization are used). Typically,  $l$  is less than  $N$ .

Motion is defined by a time dependent constraint equation (*driving constraints*):

$$\Phi^d(\mathbf{q}, t) = \mathbf{0}. \quad (2)$$

In the most general case, when constraint equations are induced by either joints or driving constraints, the following equations must be satisfied at any time  $t$  (*position level analysis*):

$$\Phi(\mathbf{q}, t) = \begin{bmatrix} \Phi^k(\mathbf{q}, t) \\ \Phi^d(\mathbf{q}, t) \end{bmatrix} = \mathbf{0} \quad (3)$$

and (*velocity and acceleration level analysis*, respectively):

$$\frac{d\Phi(\mathbf{q}, t)}{dt} = \Phi_{\mathbf{q}}\dot{\mathbf{q}} + \Phi_t = \mathbf{0}, \quad (4)$$

$$\frac{d^2\Phi}{dt^2} = \Phi_{\mathbf{q}}\ddot{\mathbf{q}} + (\Phi_{\mathbf{q}}\dot{\mathbf{q}})_{\mathbf{q}}\dot{\mathbf{q}} + 2\Phi_{t_{\mathbf{q}}}\dot{\mathbf{q}} + \Phi_{tt} = \Phi_{\mathbf{q}}\ddot{\mathbf{q}} - \Gamma = \mathbf{0}. \quad (5)$$

If constraints (3) are independent in point  $\widetilde{\mathbf{q}}_0 = [\mathbf{q}_0^T \ t_0]^T$  (regular point), i.e. if:

$$\text{rank}(\Phi_{\mathbf{q}}) = N, \quad (6)$$

then unique solutions of linear systems (4) and (5), as well as nonlinear system (3), exist in the neighbourhood of point  $\widetilde{\mathbf{q}}_0 = [\mathbf{q}_0^T \ t_0]^T$ . From numerical point of view,

the kinematic analysis of a system described by (3) can be considered as a numerical tracing of a trajectory  $T$  such that:

$$T = \left\{ \widetilde{\mathbf{q}} = [\mathbf{q}^T, t]^T : \Phi(\widetilde{\mathbf{q}}) = \mathbf{0}, \mathbf{q} = \mathbf{q}(t), t_0 < t < t_1, \widetilde{\mathbf{q}} \in R^{N+1} \right\}. \quad (7)$$

In general, the numerical tracing of trajectories (7) is the subject of the *numerical continuation methods* [5]. One of the simplest methods is the Euler predictor-Newton corrector scheme. The classical Newton corrector can be defined in the form:

$$\Phi_{\mathbf{q}}(\mathbf{q}^k, t^{i+1})\Delta\mathbf{q}^k + \Phi(\mathbf{q}^k, t^{i+1}) = \mathbf{0}. \quad (8)$$

The iterative algorithm (8) is numerically very efficient, however, under strong assumption that condition (6) is fulfilled (i.e., all points of the trajectory are regular) and when good starting position is chosen. In case the trajectory contains singular points which correspond to robot singular positions, Eq. (8) becomes ill-conditioned and simulation usually fails.

We will distinguish two simplest cases of singular configuration of MBS [8]:

1. *Turning point (limit point or fold bifurcation)* – when  $\text{rank}(\Phi_{\mathbf{q}}) = N - 1$  and  $\text{rank}(\Phi_{\widetilde{\mathbf{q}}}) = N$  and there exists a parameterization  $\mathbf{q}(\tau)$ ,  $t(\tau)$ , with  $\mathbf{q}(\tau_0) = \mathbf{q}_0$  and  $t(\tau_0) = t_0$ , and  $d^2t/d\tau^2 \neq 0$ .
2. *Simple stationary bifurcation point* – when  $\text{rank}(\Phi_{\mathbf{q}}) = \text{rank}(\Phi_{\widetilde{\mathbf{q}}}) = N - 1$  and exactly two branches of solutions intersect with two distinct tangents.

In both cases point  $\widetilde{\mathbf{q}}_0 = [\mathbf{q}_0^T, t_0]^T$  fulfils Eq. (7).

The geometrical representations of these two singular configurations are shown in Fig. 1 on the example of a slider-crank mechanism.

In order to numerically detect and describe two simple cases of singular configuration, general continuation scheme was applied [5]. The Newton corrector (8) was replaced with a more general corrector given by:

$$\widetilde{\mathbf{q}}^{k+1} = \widetilde{\mathbf{q}}^k - \Phi_{\mathbf{q}}^+(\widetilde{\mathbf{q}}^k)\Phi(\widetilde{\mathbf{q}}^k) \text{ and } \widetilde{\mathbf{q}}^0 = \widetilde{\mathbf{q}}^{i+1}, \quad (9)$$

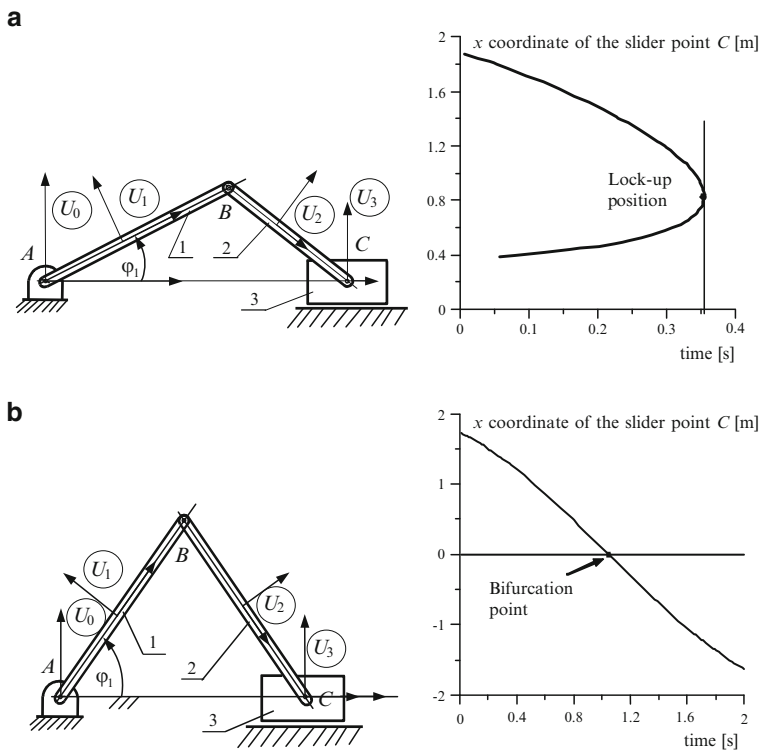
where  $(.)^+$  denotes pseudo-inverse matrix (Moore–Penrose).

Pseudo-inverse matrix can be calculated efficiently using sparse matrix  $\mathbf{J}^{-1}$  given by the formula:

$$\mathbf{J}^{-1} = \begin{bmatrix} \Phi_{\mathbf{q}} & \Phi_t \\ \mathbf{e}^T & \chi \end{bmatrix}^{-1} = \begin{bmatrix} \Phi_{\mathbf{q}}^{-1} + \Phi_{\mathbf{q}}^{-1}\Phi_t s^{-1}\mathbf{e}^T\Phi_{\mathbf{q}}^{-1} & -\Phi_{\mathbf{q}}^{-1}\Phi_t s^{-1} \\ -s^{-1}\mathbf{e}^T\Phi_{\mathbf{q}}^{-1} & s^{-1} \end{bmatrix}, \quad (10)$$

$$s = \chi - \mathbf{e}^T\Phi_{\mathbf{q}}^{-1}\Phi_t.$$

In order to trace trajectory (branch) numerically, local parameterization strategy was chosen [5]. Instead of the time variable  $t$ , other parameter is chosen as the independent one. Symbol  $\chi$  in formula (10) is responsible for the parameter choice.



**Fig. 1** Slider crank mechanism (planar example). Driving constraints are determined by equation:  $\varphi_1 - \omega t = 0$ : (a)  $AB/BC > 1$  turning point (lock-up position), (b)  $AC = BC$  (simple stationary bifurcation point). In both cases singular position occurs when link  $BC$  reaches vertical position

The independent parameters are constant in the intervals of time. Moore–Penrose matrix can be calculated efficiently with the formula:

$$\Phi_q^+ = (\mathbf{I} - \mathbf{s}\mathbf{s}^T)(\mathbf{J}^{-1})_N, \quad (11)$$

where  $\mathbf{s}$  is a tangent vector and  $(\mathbf{J}^{-1})_N$  denotes the submatrix of matrix (10) built of the first  $N$  columns. It should be pointed out that, in the algorithm of matrix  $\mathbf{J}^{-1}$  evaluation, sparsity of the matrices can be exploited intensively using, e.g., well known subroutines for sparse matrices.

For bifurcation point detection a test function  $\zeta$  is introduced, which is evaluated during the branch tracing. A bifurcation is indicated by function  $\zeta$  value – at the bifurcation point the test function satisfies condition:  $\zeta = 0$ . For the branch tracing given by iterative scheme (10), the test function can be proposed in the form:

$\zeta = \det \begin{pmatrix} \Phi_q \\ \tilde{\mathbf{J}}^T \end{pmatrix}$ . This expression can be evaluated very efficiently if sparsity of the Jacobian matrix  $\Phi_q$  is taken into account.

In the close neighbourhood of the bifurcation point singular position can be evaluated with greater accuracy using direct method for branch calculating. The equation set can be extended to the new branch system:

$$\tilde{\Phi}(\mathbf{Y}) = \begin{bmatrix} \Phi(\mathbf{q}, t) \\ \Phi_{\mathbf{q}}(\mathbf{q}, t) \mathbf{h} \\ h_k - 1 \end{bmatrix} = \mathbf{0}, \quad \mathbf{Y} = [\mathbf{q}^T \ t \ \mathbf{h}^T]^T, \quad (12)$$

where  $\mathbf{h}$  is a tangent vector.

System (12) can be solved using efficient numerical solvers for sparse matrix equation and Newton-like iterative schemes.

It should be pointed out that presented algorithms can be used for branch switching, i.e., calculating one (at least) solution on the emanating branch. If one of the solutions is situated somewhat close to the bifurcation point, then other solution on the emanating branch can be found using techniques of perturbations, widely used in numerical continuation theory.

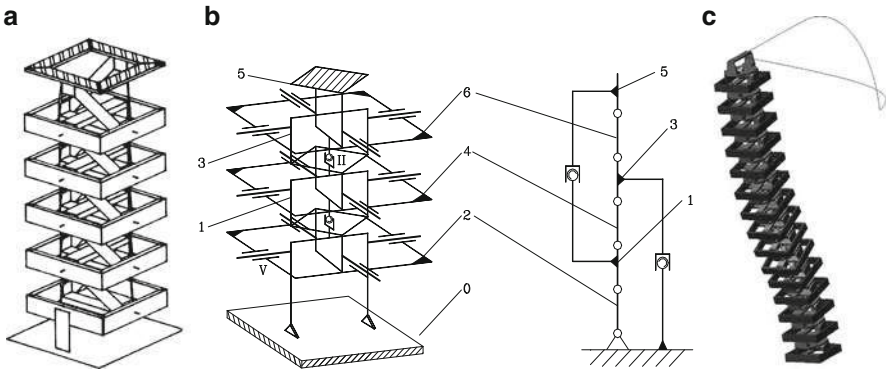
A general numerical test program was developed using the described above general techniques for branch tracing and switching. Its idea was based on the research package named BIFPACK [8]. Numerical algorithm used in the test program can be implemented also in general environment of multibody codes, as a separate module devoted to branch tracing and simple singularity diagnosis in kinematic analyses.

## 2.2 Robotic Example

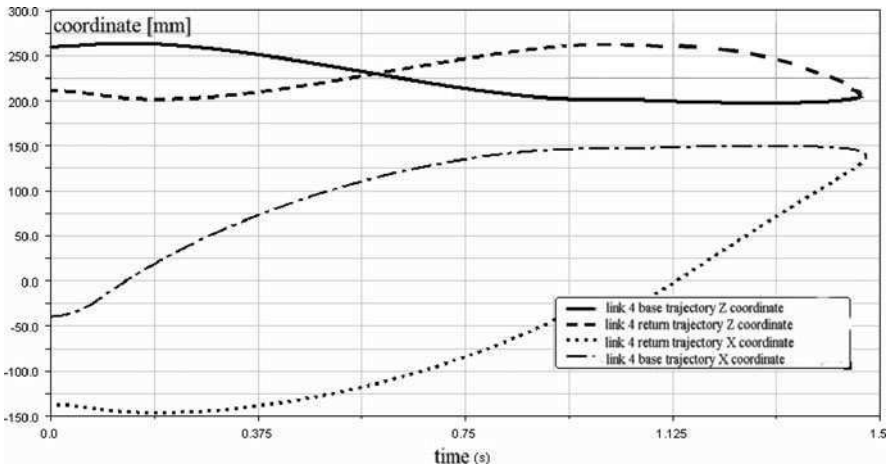
Practical applications of developed ideas can be demonstrated on the example of a multilink robot [9], which was primarily intended to weld car body in places which are not easy to reach. The robot consists of several sets of bodies, called segments (Fig. 2b). Segments are built of rigid parts. Kinematic scheme of an exemplary segment is presented in Fig. 2b. Every segment consists of  $4 + 2n$  rigid parts connected by spherical–translational and revolute joints. The segment mobility (Grubler count) does not depend on number  $n$  and is always equal to 2. Kinematic constraints imposed on bodies are independent (there are no redundant constraints). Two, three or four (generally  $m$ ) segments with different or equal number of bodies can be connected, giving a manipulator with Grubler count equal to 4, 6 and 8 ( $2m$ ), respectively. In Fig. 2c a multilink robot built of three segments, with ten bodies each, is presented.

Various analyses were preformed during the dimensional and structural synthesis process, and the following problems were investigated:

- Deciding whether the desired trajectory can be realised for given structures and dimensions of the robot (detection of lock-up positions)
- Detection of singular positions of the robot, and particularly positions where kinematic parameters of robot links are not continuous functions of time



**Fig. 2** Multilink robot: (a) One segment of consisting of 12 rigid parts, (b) kinematic scheme of a six-part segment, (c) robot built of three 10-part segments



**Fig. 3** Global coordinates (x, z) of the multilink robot top part centre

- Determining the relative angles in the actuated revolute joints as functions of time (inverse kinematics)

The robot kinematics was simulated using a model built in a general multibody package. Absolute coordinate formulation was used and test functions were implemented.

In Fig. 3 one of the simulation results is shown. The turning point is detected, which can be geometrically interpreted as a lock-up position of the mechanism. Full analysis of robot behaviour in the neighbourhood of turning point was possible, since the branch tracing program was used.



### **3 Simulation Study of Stewart Platform with Model-Based Control**

Robot is a complex system consisting of manipulator mechanisms, actuators, sensors and control unit electronics. Due to the system complexity it is usually difficult to predict newly designed robot behaviour without numerical simulation. Thus building a numerical model quite often becomes necessary to provide the robot designers with required data. Each subsystem of the robot can be analyzed separately, however, the best results are obtained when all subsystems are modelled and simulated jointly. The general multibody methods and software can be relatively easily used to model the robot mechanisms, whereas other robot subsystems usually need different methods and software to be simulated [10, 11]. The multibody methods and models are versatile and thus they can be linked with other methods, which provide a good platform for integration of various scientific and engineering disciplines.

In this section a simulation study of parallel robot and its control system is presented. Two well-known software packages, one designed to perform control system simulation and the other dedicated to multibody simulation, have been used to conduct the study. These packages were cooperating during simulations – both programs performed all calculations simultaneously.

#### **3.1 Methods**

In contemporary parallel robots the position control – widely used in earlier designs – is replaced by a model-based control. Due to high sampling frequency of control systems, the driving forces calculations must be performed fast. Therefore the inverse dynamics model of manipulator is usually simplified, and thus some aspects of manipulator motion are not represented in their full complexity.

A 6-dof Stewart platform type parallel manipulator was investigated. The investigated control scheme employed the dynamics of platform and actuators. Friction was included in the model. The aim of the study was to check what is the influence of dynamics model accuracy on the quality of control process. The simplifications of the model and the problems with finding accurate values of its parameters (especially friction parameters) were considered. The results of this study helped to decide how big could be the simplifications of the dynamics model and to predict what are the possible results of inaccurate determination of the crucial model parameters. The obtained results were also helpful when controller parameters were searched for.

The manipulator forward dynamics was modelled using a multibody package, which automatically generates and solves the equations of motion. Therefore it was relatively easy to introduce changes into the model and to take into account various factors, for example joint friction or interactions with environment. There was no need for tedious and difficult process of deriving and programming manually the

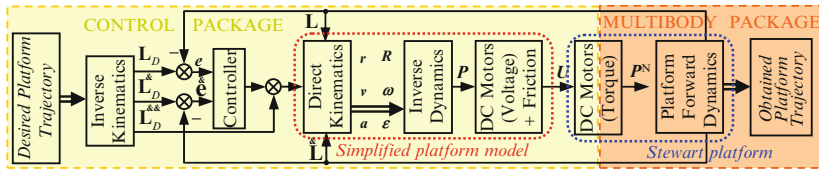


Fig. 4 Schematic view of the model

necessary equations. Therefore, crude simplifications of the multibody model were not required. The additional benefit of using multibody package was the possibility to create and watch animations of the manipulator in motion.

The control system and the electric actuators were modelled in the control software simulation package. Since the model-based control scheme was adopted, the inverse dynamics problem had to be solved within the control system. Friction effects were included in the inverse dynamics model. In order to enable fast calculations, the model utilized by the control system was simplified.

The model scheme is shown in Fig. 4. It is worth noting that electric phenomena in actuators were included in the control package part of the model, whereas the mechanics of actuators (inertial properties, gearing etc.) was modelled in the multibody package.

### 3.1.1 Manipulator Kinematics

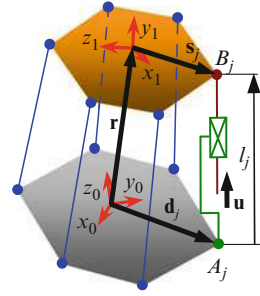
A kinematic scheme of the manipulator is presented in Fig. 5. To simplify the picture, details of only one leg are presented. The coordinates of position vectors  $\mathbf{d}_j$  ( $j = 1, \dots, 6$ ) are constant in the  $\pi_0$  frame (established on the manipulator basis), whereas coordinates of position vectors  $\mathbf{s}_j^{(1)}$  ( $j = 1, \dots, 6$ ) are constant in the  $\pi_1$  frame (established on moving platform).

The kinematics of moving platform is described by absolute coordinates. The position of local frame  $\pi_1$  in the global frame  $\pi_0$  is described by vector  $\mathbf{r}$ , and the orientation of  $\pi_1$  frame with respect to  $\pi_0$  frame is given by three Euler ( $z-x'-z''$ ) angles:  $\varphi_1, \varphi_2, \varphi_3$ . The coordinates of vector  $\mathbf{r}$  and angles  $\varphi_1, \varphi_2, \varphi_3$  are assumed functions of time.

The *inverse kinematics problem* consists in searching for the actuators motion (lengths, velocities and accelerations) when the platform motion is given (position, velocity, acceleration). Since in the inverse kinematics problem position vector  $\mathbf{r}$  and direction cosines matrix  $\mathbf{R}$  are given, vector  $\mathbf{l}_j$  from point  $A_j$  to point  $B_{Bj}$ , distance  $l_j$  and unit vector  $\mathbf{u}_j$  (see Fig. 5) can be calculated as:

$$\mathbf{l}_j = \mathbf{r} + \mathbf{R}\mathbf{s}_j^{(1)} - \mathbf{d}_j, \quad l_j = \sqrt{\mathbf{l}_j^T \mathbf{l}_j}, \quad \mathbf{u}_j = \mathbf{l}_j / l_j. \quad (13)$$

**Fig. 5** Simplified kinematic scheme of the manipulator



Differentiation of the above formulas leads to the velocity equation:

$$\dot{l}_j = \mathbf{u}_j^T \dot{\mathbf{l}}_j = \mathbf{u}_j^T (\mathbf{v} + \tilde{\boldsymbol{\omega}} \mathbf{s}_j) = \mathbf{u}_j^T \mathbf{v} - \mathbf{u}_j^T \tilde{\mathbf{s}}_j \boldsymbol{\omega} = \mathbf{J}_j \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix}. \quad (14)$$

In the above equation  $\mathbf{J}_j$  denotes  $j$ th row of manipulator Jacobian matrix:

$$\mathbf{J}_j = \begin{bmatrix} \mathbf{u}_j^T & -\mathbf{u}_j^T \tilde{\mathbf{s}}_j \end{bmatrix}, \quad (15)$$

$\mathbf{v}$  and  $\boldsymbol{\omega}$  are linear and angular velocities, respectively, and symbol  $\tilde{\boldsymbol{\omega}}$  denotes the skew-symmetric matrix associated with vector  $\boldsymbol{\omega}$ .

Similarly, in the case of inverse kinematics, the acceleration equation can be formulated in a closed form [12]. Finally, actuator lengths, velocities and accelerations can be grouped into six-component-vectors:

$$\mathbf{L} = [l_1 \cdots l_6]^T, \quad \dot{\mathbf{L}} = [\dot{l}_1 \cdots \dot{l}_6]^T, \quad \ddot{\mathbf{L}} = [\ddot{l}_1 \cdots \ddot{l}_6]^T. \quad (16)$$

The *direct kinematics problem* consists in searching for the platform motion (position, velocity, acceleration) when the actuators motion is given (lengths, velocities and accelerations). Equations (13) can be reformulated to obtain:

$$\Phi_j(\mathbf{r}, \boldsymbol{\varphi}) \equiv \left( \mathbf{r} + \mathbf{R}(\boldsymbol{\varphi}) \mathbf{s}_j^{(1)} - \mathbf{d}_j \right)^T \left( \mathbf{r} + \mathbf{R}(\boldsymbol{\varphi}) \mathbf{s}_j^{(1)} - \mathbf{d}_j \right) - l_j^2 = 0, \quad (17)$$

where  $\boldsymbol{\varphi} = [\varphi_1 \ \varphi_2 \ \varphi_3]^T$ .

The above equation can be formulated for each of the six legs, thus a set of six nonlinear algebraic equations ( $\mathbf{r}$  and  $\boldsymbol{\varphi}$  are the unknown parameters) is obtained. During direct kinematics calculations the equations are solved using the iterative Newton–Raphson method. Several solutions can be found, however, we are interested only in this one which corresponds to the admissible configuration of the manipulator. That is why the initial guess  $\mathbf{q}^0$  should be chosen carefully. It was found that good results are obtained when the iterations start from point  $\mathbf{q}^0$  which represents the central point of the manipulator workspace [12]. Some numerical tests have proven that iterations converge to the proper solution.

Calculation of unknown position ( $\mathbf{r}$  and  $\boldsymbol{\varphi}$ ) is followed by calculation of velocity ( $\mathbf{v}$  and  $\boldsymbol{\omega}$ ) and then – acceleration ( $\mathbf{a}$  and  $\boldsymbol{\epsilon}$ ). Linear equations set are solved in the case of velocity and acceleration problems.

### 3.1.2 Manipulator Dynamics

The *forward dynamics problem* consists in searching for the mechanism motion, when forces actuating the mechanism are known. The forward dynamics of manipulator was modelled using multibody package. This program automatically generates and solves multibody system equations of motion. Thus, there is no need to derive the motion equations in a full (i.e. not simplified) form.

The inverse problem of dynamics consists in searching for driving forces, which are necessary to obtain the desired motion of mechanism. The manipulator control system employs the simplified inverse dynamics model. To simplify calculations, it was assumed that all the parts of mechanism, except for the moving platform, are massless. Moreover, friction in joints was neglected (the only exception was friction in the linear actuators, which is described in the subsequent text).

It was assumed that platform centre of mass coincides with the origin of  $\pi_1$  frame. The platform is characterized by mass  $m$  and inertia matrix  $\mathbf{I}^{(1)}$  with constant elements calculated with respect to the local (i.e., moving with the platform) frame  $\pi_1$ . The inertia properties calculated with respect to the centre of mass and axes parallel to the global frame  $\pi_0$  are not constant and depend on the platform instantaneous orientation. The inertia matrix (with respect to axes parallel to  $\pi_0$  frame) can be calculated using the following equation:

$$\mathbf{I} = \mathbf{R} \mathbf{I}^{(1)} \mathbf{R}^T. \quad (18)$$

The Newton law relates the total force acting on the platform with the platform mass and center of mass linear acceleration:

$$\mathbf{F} = m\mathbf{a}. \quad (19)$$

The Euler equation relates the total torque about the platform centre of mass with the platform angular velocity, acceleration and inertia matrix:

$$\mathbf{M} = \mathbf{I}\boldsymbol{\epsilon} + \tilde{\boldsymbol{\omega}} \mathbf{I} \boldsymbol{\omega}. \quad (20)$$

For assumed platform motion, force  $\mathbf{F}$  and torque  $\mathbf{M}$  can be calculated directly from Eqs. (19) and (20).

The manipulator Jacobian matrix  $\mathbf{J}$  relates forces  $P_j$  developed by actuators to total force  $\mathbf{F}$  and total torque  $\mathbf{M}$  [13, 14]:

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} = -\mathbf{J}^T \mathbf{P}, \quad \mathbf{P} = [P_1 \dots P_6]^T. \quad (21)$$

To solve the (simplified) inverse problem of dynamics it is sufficient to perform calculations according to Eqs.(18)–(20) and then to solve the set of linear equations (21).

### 3.1.3 Friction in Actuators

The moving platform is driven by six linear actuators. Ball-screw mechanism transforms rotational motion of a DC motor into linear motion of the actuator, thus force–torque relation is the following:

$$P_j^N = T_j / h, \quad (22)$$

where  $P_j^N$  is the nominal force developed by actuator,  $T_j$  is the resultant electromagnetic torque applied to DC motor armature and  $h$  is a constant value that depends on actuator ball-screw pitch of thread and on motor gearing ratio.

The actuator output force  $P_j$  differs from nominal force  $P_j^N$ . The difference is caused by the presence of friction effects in actuator. The following model of  $j$ th actuator friction force  $P_j^F$  was used by the control package during simulations:

$$P_j^F = \begin{cases} F_C \operatorname{sgn}(\dot{l}_j) + b\dot{l}_j & \dot{l}_j \neq 0 \\ P_j^{ext} & \dot{l}_j = 0, \left| P_j^{ext} \right| < F_S \\ F_S \operatorname{sgn}(P_j^{ext}) & \dot{l}_j = 0, \left| P_j^{ext} \right| \geq F_S \end{cases}, \quad (23)$$

where  $b$  is the viscous friction coefficient,  $F_C$  is the Coulomb friction force,  $F_S$  is the maximal stiction force, and  $P_j^{ext}$  is the external force.

### 3.1.4 DC Motor

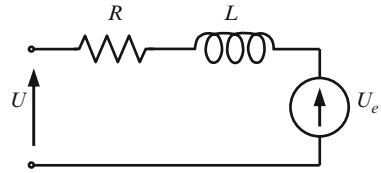
Permanent magnet DC motors are used to actuate the manipulator. Mechanical part of the motor, i.e., stator, rotor and integrated gearbox are modelled in the multibody software (which automatically formulates and solves equations), whereas electro-dynamics of motor is modelled in the control software (where the user must formulate equations).

A simplified DC motor circuit diagram is presented in Fig. 6. Parameters  $R$  and  $L$  represent resistance and inductance of armature winding, respectively.  $U_e$  represents the back-emf (electromotive force) which is generated when rotor revolves. Thus, the following equation of electric circuit can be formulated [15]:

$$U(t) = U_e(t) + L\dot{I}(t) + RI(t), \quad (24)$$

where  $U$  is the voltage of power supply,  $I$  is the armature current and  $t$  denotes time.

**Fig. 6** Simplified circuit diagram of permanent magnet DC motor



The back-emf is proportional to the motor angular velocity  $\omega$  and the torque generated by the motor is proportional to the current through the windings, hence:

$$U_e = K \cdot \omega(t), \quad T = K \cdot I(t), \quad (25)$$

where  $K$  is the motor constant (the same for both equations).

Equations (24) and (25) can be combined to obtain:

$$\dot{I} = (U - K_e \omega - RI)/L. \quad (26)$$

The above equation is integrated during direct dynamics calculations (for each motor). Then Eq. (25) is used to driving torque calculation. The calculated torque  $T_j$  is sent to multibody software and the instantaneous motor angular velocity  $\omega_j$  is received from this software.

During inverse dynamics calculations the desired torque is calculated first:

$$T_j = J \ddot{l}_j 2\pi / h + P_j^N h. \quad (27)$$

where  $J$  is the rotor, gearbox and screw reduced moment of inertia and  $P_j^N$  is defined as:

$$P_j^N = \begin{cases} P_j + P_j^F \operatorname{sgn}(\dot{l}_j) & \dot{l}_j \neq 0 \\ 0 & \dot{l}_j = 0, \left| P_j^{ext} \right| < P_j^F \\ P_j - P_j^F & \dot{l}_j = 0, \left| P_j^{ext} \right| \geq P_j^F \end{cases}, \quad (28)$$

while  $P_j$  is the force obtained by solving linear equations set (21).

Next, Eq. (25) is employed to calculate the armature current, and finally Eq. (24) is used to calculate the required control voltage (finite difference method is used to estimate the derivative of current).

### 3.1.5 Control System

The control system scheme is presented in Fig. 4. The control law is designed to reduce the control errors on position and velocity levels simultaneously. The control currents are calculated to satisfy the following error dynamics equation:

$$\ddot{\mathbf{e}} + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = \mathbf{0}, \quad (29)$$

where  $\mathbf{e} = \mathbf{L}_D - \mathbf{L}$  is the position error ( $\mathbf{L}_D$  is a six-element vector of desired actuator lengths),  $\mathbf{K}_p = k_p \mathbf{I}_{6 \times 6}$  and  $\mathbf{K}_v = k_v \mathbf{I}_{6 \times 6}$  are diagonal matrices of control gains. The gain coefficients  $k_p$  and  $k_v$  are selected to achieve the critical damping of the system described by Eq. (29).

At the beginning of computation the required motion of the platform is calculated. Then the inverse kinematics problem is solved to find the actuators' desired lengths  $\mathbf{L}_D$ , velocities  $\dot{\mathbf{L}}_D$  and accelerations  $\ddot{\mathbf{L}}_D$ .

In the real manipulator the actual actuators lengths  $\mathbf{L}$  and velocities  $\dot{\mathbf{L}}$  are measured by appropriate sensors. In the simulation model these values are computed by the multibody package, to provide feedback for the control system model.

The next step of computations consists in accelerations calculation. For given vectors of  $\mathbf{L}_D$ ,  $\dot{\mathbf{L}}_D$ ,  $\ddot{\mathbf{L}}_D$ ,  $\mathbf{L}$  and  $\dot{\mathbf{L}}$ , the vector of accelerations  $\ddot{\mathbf{L}}$ , which satisfies Eq. (29), is calculated:

$$\ddot{\mathbf{L}} = \ddot{\mathbf{L}}_D + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = \ddot{\mathbf{L}}_D + \mathbf{K}_v (\dot{\mathbf{L}}_D - \dot{\mathbf{L}}) + \mathbf{K}_p (\mathbf{L}_D - \mathbf{L}). \quad (30)$$

Then the inverse problem of dynamics is solved. The driving forces necessary to produce the required motion (described by  $\mathbf{L}_D$ ,  $\dot{\mathbf{L}}_D$  and  $\ddot{\mathbf{L}}_D$ ) are calculated. Friction forces in actuators are taken into account. The inverse dynamics calculations must be preceded by the robot direct kinematic solution, to obtain the platform motion. The last step of computations consists in calculation of control voltage for all actuators.

### 3.2 Results

The presented simulation model of a parallel manipulator, with ball-screw actuators driven by DC motors and with model-based control system, enables to analyze various problems concerning the system behaviour. The simulation results provide the robot designer with data necessary to make decisions. Since the model-based control is employed, this particular study was focused on checking what is the influence of dynamics model simplifications and model parameter uncertainties on the quality of control process.

The actuator masses are neglected in the inverse dynamics model, thus the moving platform is the only system element with non-zero mass. In the inverse dynamics model, which is employed by the control system, the actuator masses may be considered in the simplified way, by appropriate enlargement of the moving platform mass. The obtained results showed that inverse dynamics model simplifications consisting in neglecting the mass of actuators have relatively little influence on position errors and even smaller influence on velocity errors. It was also found, that the effects of actuators masses neglecting can be significantly reduced by appropriate changes in the modelled mass of the platform. Thus, there is no need to use more accurate (and requiring much more computation) inverse dynamics model.

Employing a friction model in the model-based control inverse dynamics calculations can reduce the problems caused by actuators friction. It should be stated, however, that friction is a complicated phenomenon and its computational models are usually severely simplified. Moreover, the friction parameters are usually difficult to measure and quite often are time-varying. Thus, it should be expected that the friction model utilized during the inverse dynamics calculations would not be accurate. A series of simulations was performed in order to check what is the influence of friction forces and friction model inaccuracies on the obtained quality of control. In some simulations the friction parameters used by the inverse dynamics model were different than friction parameters used by the direct dynamics model (underestimated or overestimated). It was found that friction effects should be introduced to the inverse dynamics model, since it importantly improves the quality of control. The simulation results showed that the friction parameters should be identified with big accuracy. If the friction model parameters are not accurate enough, the quality of control does not improve. It was also found during investigations that the parameters describing stiction-friction transition effects are the crucial ones.

At the end it is worth noting that the presented simulation model can be modified with small effort, thus the designer can easily investigate and assess various variants of whole robot or its subsystems.

## **4 Dynamic Analysis of a Flexible Power Transmission Mechanism**

General multibody formalism can be also used for robot modelling when flexibility effects must be taken into account. The MBS code is applied on early stage of robot synthesis and, using general purpose tools, allows predicting various characteristics of mechanisms.

The most common approach adapted in many commercial and research MBS packages is based on floating frame of reference formulation (FRF). In this approach models of flexible bodies are usually prepared using classical FEM formulations and then transferred to MBS code as substructures in various representations. Kinematic and dynamic analysis is most often performed directly in MBS program. The well known drawback of floating frame formulation is the restriction that deformations of bodies should be small (although displacements are large). Moreover, in practice substructure is represented by only dozen or so first modes, and the change of number of modes requires additional time-consuming calculations.

In this chapter we present dynamic analysis of a flexible parallelogram mechanism. This mechanism is used for power transmission in POLYCRANK robot designed in the Warsaw University of Technology [16]. A general flexible MBS method, based on FRF was used. A method of joint friction modelling that accounts for assembly stresses was also applied. Output functions, i.e. angular velocities and accelerations of selected bodies, friction and reaction forces in kinematic pairs, as



well as stress maps in selected bodies are presented. In addition, a discussion of different integration methods influence on the results accuracy is briefly presented.

All computations were carried out using CAD/CAE packages commonly used in virtual prototyping [17].

#### 4.1 An Outline of Dynamic Analysis of Flexible MBS

In classical absolute coordinates formulations an intermediate body-fixed local reference frame (called a *floating frame*) is often introduced to describe large motion and small deformations of a body. Six generalized coordinates representing large motion of the body (e.g., three Cartesian coordinates of the centre of floating frame and three Euler angles) describe local body reference frame position and orientation with respect to the global (fixed) reference frame. Body deformations are described using FEM linear theory with respect to the local reference frame. Equations of motion of the whole system may be written using, for example, Lagrange equations of the first kind in the form [18]:

$$\begin{aligned} \begin{bmatrix} \mathbf{m}_{SS} & \mathbf{m}_{Sf} \\ \mathbf{m}_{fS} & \mathbf{m}_{ff} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_S \\ \ddot{\mathbf{q}}_f \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{ff} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_S \\ \dot{\mathbf{q}}_f \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{ff} \end{bmatrix} \begin{bmatrix} \mathbf{q}_S \\ \mathbf{q}_f \end{bmatrix} = \\ = \begin{bmatrix} (\mathbf{Q}_z)_S \\ (\mathbf{Q}_z)_f \end{bmatrix} + \begin{bmatrix} (\mathbf{Q}_v)_S \\ (\mathbf{Q}_v)_f \end{bmatrix} - \Phi_{\mathbf{q}}^T \boldsymbol{\lambda}, \end{aligned} \quad (31)$$

where:  $\mathbf{q} = [\mathbf{r}^T \boldsymbol{\theta}^T \mathbf{q}_f^T]^T = [\mathbf{q}_S^T \mathbf{q}_f^T]^T$  is a vector of the rigid and flexible general coordinates,  $\mathbf{m}$  are mass matrices,  $\mathbf{D}$  is a dumping matrix (resulting from dissipation function),  $\Phi$  is a vector of constraint equations imposed by joints connecting bodies,  $\Phi_{\mathbf{q}}^T \boldsymbol{\lambda}$  is a vector of reaction forces represented by Lagrange multipliers,  $\mathbf{Q}_z$  is a vector of external forces applied to the body,  $\mathbf{Q}_v$  is a vector of centrifugal, Coriolis and other forces which result from differentiation of the kinetic energy with respect to time and coordinates.

Equations of motion (31) form a system of differential-algebraic equations (DAE) with large number of unknowns, which depends directly on the number of FEM model degrees of freedom. That is why, in general, reduction of degrees of freedom is performed with usage of the modal synthesis algorithm. One of the most common approaches in the structural dynamics is the Craig–Bampton method [19]. Using the Craig–Bampton modes system of equations (31) can be represented in the new modal coordinates:

$$\begin{aligned} \begin{bmatrix} \mathbf{m}_{SS} & \mathbf{m}_{Sf}\Psi \\ \Psi^T \mathbf{m}_{fS} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_S \\ \ddot{\mathbf{p}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{d} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_S \\ \dot{\mathbf{p}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega} \end{bmatrix} \begin{bmatrix} \mathbf{q}_S \\ \mathbf{p} \end{bmatrix} = \\ = \begin{bmatrix} (\mathbf{Q}_z)_S \\ \Psi^T (\mathbf{Q}_z)_f \end{bmatrix} + \begin{bmatrix} (\mathbf{Q}_v)_S \\ \Psi^T (\mathbf{Q}_v)_f \end{bmatrix} - \begin{bmatrix} \Phi_{\mathbf{q}_S}^T \\ \Psi^T \Phi_{\mathbf{q}_f}^T \end{bmatrix} \boldsymbol{\lambda}, \end{aligned} \quad (32)$$

where  $\Psi$  is a rectangular transformation matrix,  $\mathbf{p}$  is a new vector of coordinates obtained after coordinate transformation,  $\mathbf{I}$  is an identity matrix and  $\mathbf{d}$  is a matrix of modal damping.

It is worth noting that:

- The size of the system of equations (32) depends on the number of modal coordinates considered in transformation matrix  $\Psi$ . From numerical point of view, size of this matrix should be small. However, it cannot be chosen arbitrarily, since matrix  $\Psi$  contains information about modal content of solution.
- The system of DAE equations (32) has high differential index (equal to 3). For this kind of system many integration methods exist; it should be pointed out that effectiveness of the numerical integration of this system depends not only on the integration algorithm but also on the form of the system and method of index stabilization.

## 4.2 Power Transmission Mechanism of POLYCRANK Robot

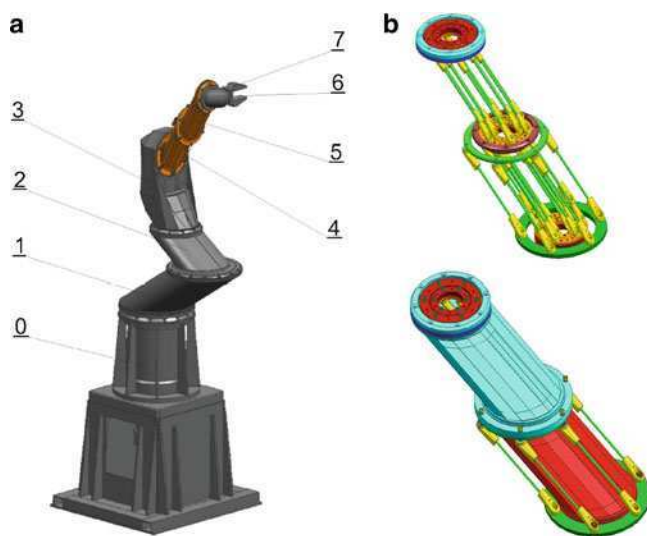
The floating reference frame approach was implemented to model fragments of power transmission mechanism of the POLYCRANK robot [16]. This robot prototype was built in Warsaw University of Technology. Unique feature of this construction is a possibility to perform unlimited rotational motion in almost all actuated joints. The robot links have the shape of diagonal cranks. Those cranks have coating construction with composite shields. Direct drive motors, situated at the robot base, are used to actuate the manipulator. The motors are connected to driven links via internal power transmission mechanisms, built of spatial parallelograms. The main parts of investigated mechanisms are shown in Fig. 7. The robot has six degrees of freedom and consists of seven links.

The complete geometrical model of the power transmission mechanism fragment, mounted into parts 4 and 5, is shown in Fig. 7b and was built in CAD environment. The power is transmitted via system of parallelograms. Each spatial parallelogram consists of:

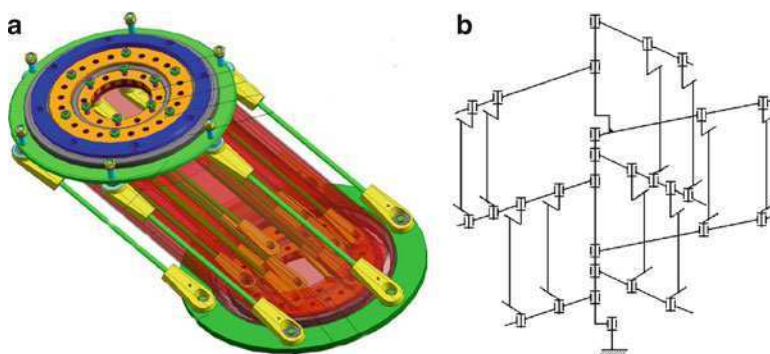
- Rotational discs, connected mutually with cross-roller bearings
- Composite shields, which serve as protection of the power transmission mechanism and make the construction stiffer
- Six connecting rods and additional elements for bearings fixing

Two parallelogram mechanisms are placed inside robot part 4 (Fig. 7a) and other two inside part 5. They transmit power from direct the drive motors to the gripper. The fifth, outer parallelogram mechanism transmits power to part 5, enabling its unlimited rotation about vertical axis.

Figure 8a illustrates part 4 with visible outer parallelogram mechanism and two inner parallelograms covered by the shield. Parallelogram linkages consist of parallel discs and rods which transmit power between discs. Each parallelogram linkage has six rods (each pull rod is ended with two heads).



**Fig. 7** POLYCRANK robot and its elements [16]: (a) General view (0 – robot base, 1, 2, 3, 4, 5, 6 – robot parts, 7 – gripper), (b) power transmission mechanism



**Fig. 8** Power transmission mechanism: (a) part 4 CAD model, (b) kinematic scheme of five parallelograms

Kinematic scheme of the five-parallelogram mechanism is presented in Fig. 8b. For simplicity, only two rods of each parallelogram linkage are considered.

A flexible MBS model of the power transmission mechanism was built and used for estimation of loads characteristics and body stresses for different variants of motion.

Due to the high complexity of robot mechanisms, some simplifications were made in the model. Only pull rods and shields were modelled as flexible. Flexibility of bearings and joints was neglected. The small parts, which do not influence on the robot dynamics, like screws or washers, were not included in the model.

### 4.3 Dynamic Analysis of Power Transmission Mechanism

Flexible models of the shield and the double-nutted pull rod were prepared using FEM package [20]. The CAD geometry models of shields were meshed using SOLID186 finite elements to obtain flexible models in FEM package (Fig. 9a). In order to connect body to other elements (discs) spider webs (beam elements) were created.

The simplified mesh of the rod built of SOLID45 elements is shown in Fig. 9b. Rigid body elements were used for simplified joint modelling (at both ends “spider webs” were created). For both types of models (shield and pull rod) substructures were created using the Craig–Bampton technique [19]. Modal content of flexible bodies was verified by comparison of modal frequencies calculated separately in FEM and in MBS [17, 20] environment.

All parts in the model were assembled using MBS techniques, to create a power transmission mechanism according to kinematic scheme presented in Fig. 8b. Additional point mass (5 kg), representing payload carried by the robot (gripper and manipulated object), was placed on the upper disc of the robot. All further calculations were carried out in MBS environment.

In the first step, a modal analysis of the whole manipulator with blocked rigid degrees of freedom was performed for various selected configurations. Knowledge about the lowest frequencies level of the mechanism natural vibrations is often important from the point of view of control system synthesis. Modal shapes corresponding to the first modal frequency (about 110 Hz) are shown in Fig. 10. Modal shape, corresponding to this frequency, represents bending of the construction along horizontal axis.

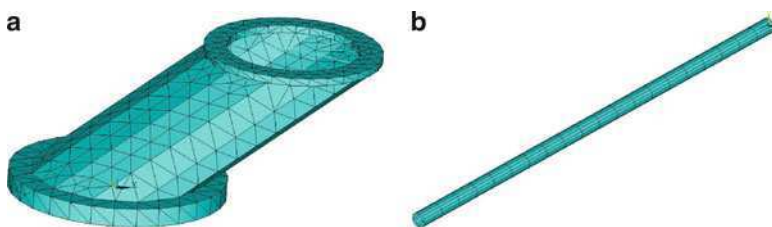


Fig. 9 FEM models: (a) part 5 shield, (b) pull rod

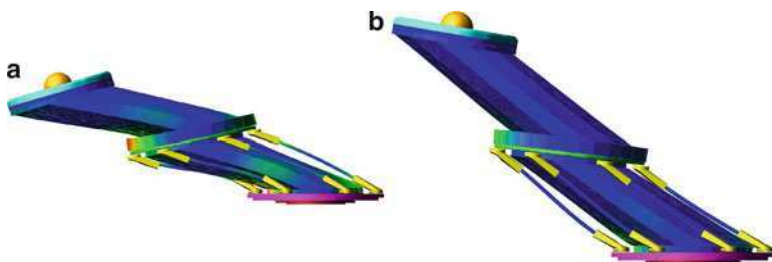


Fig. 10 Parts 4 and 5 modal shapes corresponding to the lowest modal frequency

In the next step, several dynamic analyses of the robot mechanism were performed; various time courses of driving torques were taken into consideration.

The following quantities were analyzed:

- Angles of rotations, angular velocities and accelerations of robots parts.
- Reaction forces in joints with and without friction.
- Pull rods von Mises reduced stresses.

Figures 12 and 13 show the results obtained for mechanism driven by trapezoidal input torques (presented in Fig. 11). The level of stresses in links appeared to be

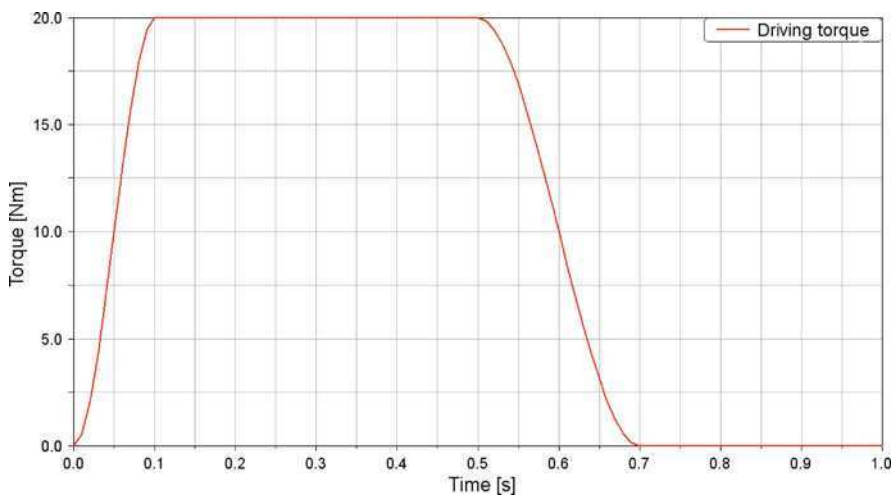


Fig. 11 Driving torque vs. time

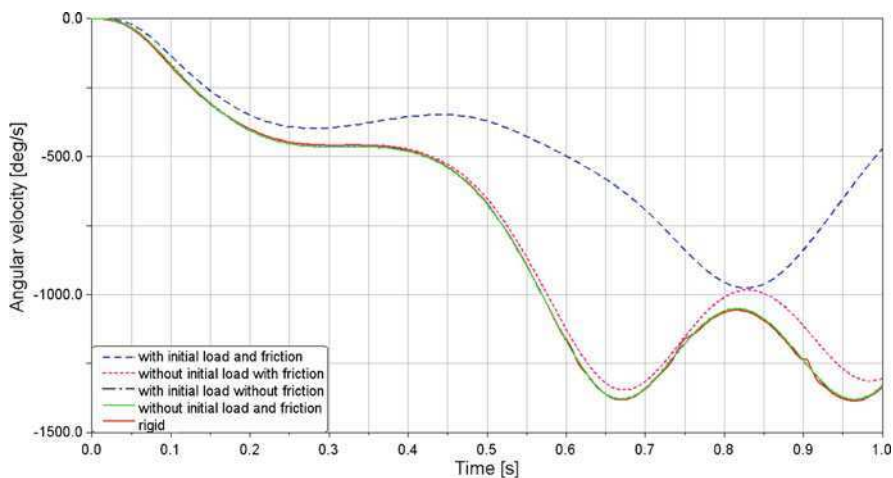


Fig. 12 Upper shield angular velocity

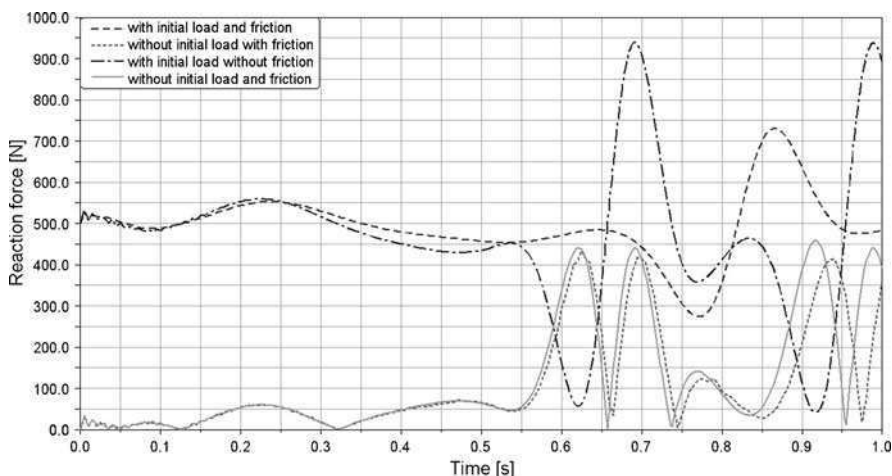


Fig. 13 Reaction forces in joint connecting cap and disc

less than several dozen of megapascal. Presence of friction and rods' preloads do not change stresses significantly, however, it can be noticed (Fig. 13) that reaction forces in joints become considerably greater when joint friction and initial loads are taken into consideration. Moreover, friction and preload result in energy losses bigger than in other cases (Fig. 12).

Finally it is worth noting that the Gear algorithm with the GGL technique of constraint stabilization was used for integration of DAE equations of motion. This algorithm was chosen on the basis of the initial tests which proved its good numerical efficiency.

## 5 Conclusions

General MBS methods can be used to estimate different robot characteristics. These methods can be useful in case of kinematic, as well as dynamic analysis. When combined with other CAE methods, the general multibody approach can be used in interdisciplinary calculations, and thus complex robotic systems can be analysed.

General multibody methods do not require closed form kinematics solution, moreover, calculations might be helpful when robot singular positions are analysed. These methods can play important role during simulation and analysis of complete robot with actuators and control system. They also become a valuable tool when effects of flexibility and joint friction should be incorporated into model.

The other important feature of general MBS formalism is that models can be easily parameterized. This makes robot performance evaluation more convenient and allows analysing various parameters. Moreover, design optimization can be performed easily from different points of view.

**Acknowledgements** The project was co-financed by the European Regional Development Fund within the framework of the 1. priority axis of the Innovative Economy Operational Programme, 2007–2013, through grant PO IG 01.02.01-00-014/08-00, and by the Institute of Aeronautics and Applied Mechanics statutory funds.

## References

1. Hunt KH (1978) Kinematic geometry of mechanism 2 and 3. Oxford University Press, Clarendon
2. Muller A (2002) Local analysis of singular configurations of open and closed loops manipulators. *Multibody Syst Dyn* 8:299–328
3. Rico JM, Gallardo JM et al (1995) A determination of singular configuration of serial nonredundant manipulators, and their escapement from singularities using lie products in computational kinematics. In: Merlet JP, Ravani B (eds) Proceedings of the 2nd workshop on computational kinematics, Sophia Antipolis, France. Kluwer, Dodrecht, The Netherlands, pp 143–152
4. Tsai LW (1999) Robot analysis. Wiley, New York
5. Allgower EL, Kurt G (1990) Numerical continuation methods, Springer series in mathematics Springer, Berlin Heidelberg
6. Frączek J (2003) Kinematic and dynamic analysis of multijoint robots using general multibody approach (a case study). In: Robot control a proceedings volume from the 7th IFAC international symposium on robot control SYROCO 2003, 1–3 Sept 2003, Wrocław, Poland
7. Nikravesh P (1988) Computer-aided analysis of mechanical systems. Prentice-Hall, Englewood Cliffs, NJ
8. Seydel R (1997) Nonlinear Computation *Int J Bifurcation Chaos* 7:2105–2126
9. Frączek J (2003) Singular position analysis of mechanism in absolute coordinates using general numerical continuation methods. In: Proceeding of the 11th world congress in MMS, China
10. Schiehlen W (1997) Multibody system dynamics: roots and perspectives. *Multibody Syst Dyn* 1:149–188
11. Bottasso CL, Riviello L (2007) Conditions for the solution of maneuvering multibody dynamics problems. In: Proceedings of the ECCOMAS thematic conference in multibody dynamics, Milan, Italy, 25–28 June 2007
12. Davliakos I, Papadopoulos E, Frączek J, Wojtyra M (2008) Simulational study of an hydraulically driven parallel manipulator control system. *J KONES Powertrain Transp* 15(1):51–68
13. Angeles J (2007) Fundamentals of robotic mechanical systems, 3rd edn. Springer Science+Business Media, New York
14. Tsai L-W (1999) Robot analysis. The mechanics of serial and parallel manipulators. Wiley, New York
15. Krause PC (1986) Analysis of electric machinery. McGraw-Hill, New York
16. Mianowski K (2001) Projektowanie i rozwój konstrukcji mechanicznych robotów mobilnych i manipulatorów (in Polish), Warsaw, Poland
17. MSC.ADAMS, 2005r2, Documentation, MSC. Software
18. Shabana AA (2005) Dynamics of multibody systems. Cambridge University Press, Cambridge
19. Craig RR, Bampton MC (1968) Coupling of substructure in dynamic analysis. *AIAA J* 6(7)
20. ANSYS rel.11, documentation, ANSYS Inc.

# Energy Considerations for the Stabilization of Constrained Mechanical Systems with Velocity Projection

Juan C. García Orden and Roberto A. Ortega Aguilera

**Abstract** There are many difficulties involved in the numerical integration of index-3 Differential Algebraic Equations (DAEs), mainly related to stability, in the context of mechanical systems. An integrator that exactly enforces the constraint at position level may produce a discrete solution that departs from the velocity and/or acceleration constraint manifolds (invariants). This behaviour affects the stability of the numerical scheme, resulting in the use of stabilization techniques based on enforcing the invariants. A coordinate projection is a poststabilization technique where the solution obtained by a suitable DAE integrator is forced back to the invariant manifolds. This paper analyzes the energy balance of a velocity projection, providing an alternative interpretation of its effect on the stability and a practical criterion for the projection matrix selection.

## 1 Introduction

Many engineering applications involve the dynamics of several bodies, rigid or deformable, undergoing large motions. Very often the motion of these systems is constrained, because there are joints that connect the different parts, or due to prescribed displacements imposed by the environment.

The mathematical models associated with these type of systems are typically formulated in terms of index-3 Differential Algebraic Equation (DAE) systems, composed of a set of differential equations, plus a set of algebraic constraint equations expressing additional relations among the generalized coordinates of the model. The numerical solution of these systems poses several difficulties, mostly related to the stability of the available integration schemes.

---

J.C.G. Orden (✉) and R.A.O. Aguilera  
ETSI Caminos, Canales y Puertos, Universidad Politécnica de Madrid,  
C/Profesor Aranguren s/n, 28040 Madrid, Spain  
e-mail: [juancarlos.garcia@upm.es](mailto:juancarlos.garcia@upm.es); [roberto.ortega@upm.es](mailto:roberto.ortega@upm.es)



Direct integration of DAEs with an index higher than one is usually not performed due to stability problems [8], although there have been some recent successful applications based on a second order generalized- $\alpha$  method applied to index-2 and index-3 DAEs (see [2] and references therein). On the other hand, index reduction through the analytical differentiation of the constraint equations causes the progressive drift of the computed solution from the position, velocity or acceleration constraint manifolds (which are invariants of the system) during the simulation. This is the point of departure of several stabilization methods found in the literature [3, 4, 6].

A coordinate projection is a poststabilization technique based on the solution of a constraint minimization problem, enforcing the solution obtained from the integrator back to the invariant's manifold. This technique has been studied and successfully applied to practical mechanical models by several researchers [1, 5, 7, 9, 10, 12, 13, 22, 25]. A detailed analysis and a discussion of the applicability of this technique can be found in the references and is beyond the scope of this paper. Nevertheless, two relevant aspects, related to the performance of this technique as applied to mechanical problems, are not found in the literature.

The first aspect is the relationship between the projection and the mechanical energy balance. It is desirable, from a physical point of view, for the behaviour of the energy of the numerical solution to be consistent with the energy of the continuous model. But this aspect is important also from the algorithmic point of view, due to the close relationship between the behaviour of the discrete energy computed by a numerical scheme and its stability [26].

The second aspect is the selection of the projection matrix. Eich [12] proposes an orthogonal projection on the null-space of the invariant, while in [1, 7, 9, 25] a mass-orthogonal projection is employed, and tested in several examples with very good results. References [10, 11] propose a projection based on the mass matrix plus other terms related to the linearized damping and elastic forces of the system, which is numerically more efficient. In fact, from a purely mathematical point of view, any positive definite matrix qualifies for a coordinate projection, which justifies the interest in searching for a practical criterion for selecting the projection matrix.

This paper focuses on these two aspects, analyzing first the energy balance involved in a coordinate projection on velocities. The results of this analysis provide an alternative interpretation of the performance of the technique, leading to a practical criterion for the matrix selection.

## 2 Constrained Dynamics Formulation

The point of departure is the formulation of the dynamics of a mechanical system with a configuration defined by the set of generalized coordinates  $\mathbf{q} \in \mathbb{R}^n$ , under the action of applied forces  $\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, t)$  and subjected to a set of  $r$  holonomic constraints  $\Phi : \mathbb{R}^n \times [0, T] \rightarrow \mathbb{R}^r$ , such that  $\Phi(\mathbf{q}, t) = \mathbf{0}$ .

The *Lagrange multiplier method* leads to an index-3 DAE system given by:

$$\mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \boldsymbol{\lambda} = \mathbf{Q}, \quad \Phi = \mathbf{0}, \quad (1)$$

$\mathbf{M}$  being the mass matrix,  $\boldsymbol{\lambda} \in \mathbb{R}^r$  the vector of Lagrange multipliers, and denoted by  $(\cdot)_{\mathbf{q}} \stackrel{\text{def}}{=} \partial(\cdot)/\partial\mathbf{q}$  and  $\dot{(\cdot)} \stackrel{\text{def}}{=} d(\cdot)/dt$ . The vector of generalized forces  $\mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, t)$  accounts for the applied forces  $\mathbf{f}$  and additional terms (gyroscopic, etc.) that may appear due to the particular type of generalized coordinates. If  $\mathbf{q}$  are Cartesian coordinates of selected points of the system, these additional terms vanish and  $\mathbf{Q} = \mathbf{f}$ .

An exact integration of the index-3 DAE system (1) in its original form (meaning that no index reduction is performed) would provide a solution that exactly satisfies the constraint at position level ( $\Phi = \mathbf{0}$ ). In this case, the constraints at velocity and acceleration levels would also be automatically exactly enforced. This means that a computed solution  $\mathbf{q}(t)$  would automatically verify  $\dot{\Phi} = \ddot{\Phi} = \mathbf{0}$ , with no further considerations; we call these *invariants* of the system.

But this situation does not hold in general for the direct numerical solution due to the approximations introduced into the computations. This means that, even though the computed solution satisfies the constraint at position level in a numerical sense (meaning that its error is below the machine precision), the solution may significantly violate the constraint at velocity and acceleration levels.

A similar situation arises when the index of the DAE system (1) is reduced by means of a double differentiation of the constraint equation, leading to the (underlying) ODE system. In this case, the numerical integration provides a solution that satisfies the constraint at acceleration level ( $\ddot{\Phi} = \mathbf{0}$ ), but progressively violates the constraints at position and velocity levels.

These facts justify the search for algorithms that force the numerical solution to remain on all the invariant's manifolds. This is the point of departure of different stabilization methods proposed in the literature; one of them is a particular poststabilization technique known as *coordinate projection*.

### 3 Coordinate Projection

With this technique, a time-stepping method is applied to (1) in order to obtain a solution for each time step, followed by a projection to bring the solution back to the invariant manifold.

In the case of a *velocity projection*, the velocities  $\dot{\mathbf{q}}^*$  computed with the integrator are projected onto the velocity constraint manifold to obtain new velocities  $\dot{\mathbf{q}}$ , solving a constrained minimization problem given by:

$$\min_{\dot{\mathbf{q}}} \frac{1}{2} (\dot{\mathbf{q}} - \dot{\mathbf{q}}^*)^T \mathbf{A} (\dot{\mathbf{q}} - \dot{\mathbf{q}}^*) \quad \text{subject to} \quad \dot{\Phi} = \mathbf{0}, \quad (2)$$

$\mathbf{A}$  being a symmetric and positive definite matrix. This minimization problem can be solved with different methods. For instance, in [25] a Lagrange multiplier method with a Newton-type iteration is employed, while in [7] an augmented-Lagrange method is used. In [10] a penalty method is used with excellent results. In the present paper we choose the same approach, a penalty method, since it allows us to obtain a closed expression for the projected velocities, while performing an efficient projection.

The penalty method transforms the constrained problem (2) into an unconstrained one, introducing a penalty parameter  $\alpha > 0$  and leading to an algebraic equation for  $\dot{\mathbf{q}}$  given by:

$$\mathbf{A}(\dot{\mathbf{q}} - \dot{\mathbf{q}}^*) + \dot{\Phi}_{\dot{\mathbf{q}}}^T \alpha \dot{\Phi} = \mathbf{0} \quad (3)$$

The terms  $\dot{\Phi}$  and  $\dot{\Phi}_{\dot{\mathbf{q}}}$  can be further elaborated as:

$$\dot{\Phi} = \frac{\partial \Phi}{\partial \mathbf{q}} \dot{\mathbf{q}} + \frac{\partial \Phi}{\partial t} = \Phi_{\mathbf{q}} \dot{\mathbf{q}} + \Phi_t; \quad \dot{\Phi}_{\dot{\mathbf{q}}} = \frac{\partial \dot{\Phi}}{\partial \dot{\mathbf{q}}} = \Phi_{\mathbf{q}}.$$

And assuming that the constraint does not explicitly depend on time ( $\Phi_t = \mathbf{0}$ ), from (3) the following linear algebraic system for the unknown  $\dot{\mathbf{q}}$  is obtained:

$$(\mathbf{A} + \alpha \Phi_{\mathbf{q}}^T \Phi_{\mathbf{q}}) \dot{\mathbf{q}} = \mathbf{A} \dot{\mathbf{q}}^* \quad (4)$$

*Remark 1.* The fact that the projection matrix  $\mathbf{A}$  is positive definite and  $\alpha > 0$  guarantees that the linear system given by (4) is non-singular, which means that the projected velocities  $\dot{\mathbf{q}}$  are always computable. In order to justify this proposition, it is only necessary to employ some standard linear algebra results, which will be used here without further proof. Recalling that  $\mathbf{A}$  is positive definite and using the fact that  $\Phi_{\mathbf{q}}^T \Phi_{\mathbf{q}}$  is positive semidefinite, the following relation holds for all  $\mathbf{x} \neq \mathbf{0}$ :

$$\mathbf{x}^T (\mathbf{A} + \alpha \Phi_{\mathbf{q}}^T \Phi_{\mathbf{q}}) \mathbf{x} = \underbrace{\mathbf{x}^T \mathbf{A} \mathbf{x}}_{> 0} + \alpha \underbrace{\mathbf{x}^T (\Phi_{\mathbf{q}}^T \Phi_{\mathbf{q}}) \mathbf{x}}_{\geq 0} > 0,$$

which means that matrix  $(\mathbf{A} + \alpha \Phi_{\mathbf{q}}^T \Phi_{\mathbf{q}})$  is positive definite and, as a consequence, it is non-singular.

Note that the use of the penalty method to solve the minimization problem (2) is approximate, in the sense that, in general, the projected velocities  $\dot{\mathbf{q}}$  do not exactly lie on the velocity constraint manifold  $\dot{\Phi}$ . In fact, it can be shown that, if a projected velocity  $\dot{\mathbf{q}}$  satisfies the velocity constraint, it is because the original velocity  $\dot{\mathbf{q}}^*$  already satisfied this constraint. This assertion is justified in the following proposition:

**Proposition 1.** *If the velocity before projection ( $\dot{\mathbf{q}}^*$ ) or the velocity after projection ( $\dot{\mathbf{q}}$ ) satisfies the velocity constraint  $\dot{\Phi} = \mathbf{0}$ , then  $\dot{\mathbf{q}}^* = \dot{\mathbf{q}}$ .*

A detailed proof of this proposition can be found in [15].

Projections can also be performed at the position and acceleration levels. For instance, in the case of an *acceleration projection*, the accelerations computed with the ODE integrator ( $\ddot{\mathbf{q}}^*$ ) are projected onto the acceleration constraint manifold to obtain new accelerations ( $\ddot{\mathbf{q}}$ ), solving a constrained minimization problem given by:

$$\min_{\ddot{\mathbf{q}}} \frac{1}{2} (\ddot{\mathbf{q}} - \ddot{\mathbf{q}}^*)^T \mathbf{A} (\ddot{\mathbf{q}} - \ddot{\mathbf{q}}^*) \quad \text{subject to} \quad \ddot{\Phi} = \mathbf{0}, \quad (5)$$

$\mathbf{A}$  being a positive definite matrix.<sup>1</sup> Again, this constrained minimization problem can be solved with penalty, which leads to the solution for  $\ddot{\mathbf{q}}$  as a linear algebraic system given by:

$$(\mathbf{A} + \alpha \Phi_q^T \Phi_q) \ddot{\mathbf{q}} = \mathbf{A} \ddot{\mathbf{q}}^* - \alpha \Phi_q^T \dot{\Phi}_q \dot{\mathbf{q}} \quad (6)$$

The analysis of position and acceleration projections is outside the scope of this paper, and only a *velocity projection* will be considered. This is justified by the results reported in [1, 22, 25]. These authors show that errors in the velocity constraint are more critical for the numerical solution than errors in the position constraint, coming to the conclusion that velocity projection is the most efficient projection for improving numerical integration.

## 4 Total Energy Balance

For systems of ODEs arising from the dynamics of mechanical systems, the stability of the numerical methods used to solve them is often related to the concept of energy. Actually, in the linear case, exact algorithmic energy conservation leads to unconditional stability, as happens, for instance, with the trapezoidal rule [23]. However, this direct relationship does not hold for the nonlinear case [24, 26], which is the case of the equations resulting from practical multibody systems. Nevertheless, exact conservation of energy (or unconditional energy dissipation) has revealed itself to be extremely useful in the design of robust integration schemes, with excellent stability in the nonlinear case ([26] and references therein) and applied to the dynamics of multibody systems [16–18, 20].

With these arguments in mind, it is interesting to analyze how the coordinate projection behaves in terms of energy balance. As will be shown below, it turns out that the projection actually controls the energy, therefore providing a new point of view for the understanding of its stabilization properties.

In order to establish a suitable point of departure, let us consider a constrained mechanical system, represented by a set of coordinates  $\mathbf{q} \in \mathbb{R}^n$ , subjected to a set of  $r$  holonomic constraints  $\Phi(\mathbf{q}) \in \mathbb{R}^r$  and without applied forces. The dynamics of this system are represented by the index-3 DAE:

$$\mathbf{M} \ddot{\mathbf{q}} + \mathbf{Q}_\Phi(\mathbf{q}) = \mathbf{Q}, \quad \Phi = \mathbf{0} \quad (7)$$

---

<sup>1</sup> Not necessarily the same employed for the velocity projection.

$\mathbf{Q}_\Phi$  being the constraint force vector, which in the case of the Lagrange multiplier method is given by  $\mathbf{Q}_\Phi = \Phi_{\mathbf{q}}^T \boldsymbol{\lambda}$ . The generalized force vector  $\mathbf{Q}$  vanishes if  $\mathbf{q}$  contains Cartesian coordinates of selected points of the system.

*Remark 2.* The fact that no applied forces (e.g. external loads or internal forces in discretized deformable bodies) are considered in (7) does not limit the applicability of the developments presented in the next sections. This is due to the fact that the velocity projection does not affect the work performed by these forces, which typically depends only on positions.

*Remark 3.* The dynamical system represented by (7) is conservative (the total mechanical energy remains constant), since the work performed by the holonomic constraints which do not depend explicitly on time is zero.

Directly integrating the index-3 DAE (7) from  $t_n$  to  $t_{n+1}$  provides a solution  $\mathbf{q}_{n+1}$  that exactly satisfies the position constraint. In consequence, the constraint force at  $t_{n+1}$  takes the value  $\mathbf{Q}_{\Phi_{n+1}} = \Phi_{\mathbf{q}_{n+1}}^T \boldsymbol{\lambda}_{n+1}$ ,  $\boldsymbol{\lambda}_{n+1}$  being the vector of exact Lagrange multipliers.

A velocity vector  $\dot{\mathbf{q}}_{n+1}^*$  is also obtained, but in general, the velocity constraint  $\dot{\Phi}_{n+1}$  is not exactly satisfied. In order to move the solution back to the velocity constraint manifold, let us assume that a velocity projection is performed at the end of each time step as explained in Sect. 3, obtaining a new velocity vector  $\dot{\mathbf{q}}_{n+1}$ .

The total discrete energy balance  $\Delta E$  between  $t_n$  and  $t_{n+1}$  is given by:

$$\Delta E = \frac{1}{2} \dot{\mathbf{q}}_{n+1}^T \mathbf{M} \dot{\mathbf{q}}_{n+1} - \frac{1}{2} \dot{\mathbf{q}}_n^T \mathbf{M} \dot{\mathbf{q}}_n \quad (8)$$

Note that the energy balance  $\Delta E$  given by (8) equals the kinetic energy balance. This is due to the fact that there are no applied forces, the position constraints are exactly satisfied, and the position  $\mathbf{q}_{n+1}$  does not change under the projection.

Adding and subtracting a term  $(1/2) \dot{\mathbf{q}}_{n+1}^{*T} \mathbf{M} \dot{\mathbf{q}}_{n+1}^*$  in (8), the following relation is obtained:

$$\Delta E = \underbrace{\frac{1}{2} \dot{\mathbf{q}}_{n+1}^{*T} \mathbf{M} \dot{\mathbf{q}}_{n+1}^* - \frac{1}{2} \dot{\mathbf{q}}_n^T \mathbf{M} \dot{\mathbf{q}}_n}_{\Delta E_i} + \underbrace{\frac{1}{2} \dot{\mathbf{q}}_{n+1}^T \mathbf{M} \dot{\mathbf{q}}_{n+1} - \frac{1}{2} \dot{\mathbf{q}}_{n+1}^{*T} \mathbf{M} \dot{\mathbf{q}}_{n+1}^*}_{\Delta E_p}, \quad (9)$$

$\Delta E_i$  being the energy variation introduced by the ODE integrator, and  $\Delta E_p$  the energy variation introduced by the velocity projection.

It is not difficult to obtain an expression for the energy variation  $\Delta E_i$  introduced by a standard ODE integrator. The point of departure is the first term of (9) rewritten as:

$$\Delta E_i = \frac{1}{2} (\dot{\mathbf{q}}_{n+1}^* + \dot{\mathbf{q}}_n)^T \mathbf{M} (\dot{\mathbf{q}}_{n+1}^* - \dot{\mathbf{q}}_n) \quad (10)$$

and using the algorithmic expressions of the method with the original system (7). For instance, for the *trapezoidal rule* the following relations hold:

$$\begin{aligned}\dot{\mathbf{q}}_{n+1}^* + \dot{\mathbf{q}}_n &= \frac{2}{\Delta t}(\mathbf{q}_{n+1} - \mathbf{q}_n) \\ \dot{\mathbf{q}}_{n+1}^* - \dot{\mathbf{q}}_n &= -\frac{\Delta t}{2}\mathbf{M}^{-1}(\mathbf{Q}_{T_n} + \mathbf{Q}_{T_{n+1}})\end{aligned}$$

with  $\mathbf{Q}_T = \mathbf{Q}_\Phi - \mathbf{Q}$ , which introduced into expression (10) give, after some algebra:

$$\Delta E_i = -(\mathbf{q}_{n+1} - \mathbf{q}_n)^T \overline{\mathbf{Q}}_{T_{n+\frac{1}{2}}}, \quad (11)$$

where the notation  $\overline{(\cdot)}_{n+\frac{1}{2}} \stackrel{\text{def}}{=} [(\cdot)_n + (\cdot)_{n+1}]/2$  has been employed.

Another example is the *implicit midpoint rule*, which introduces an energy variation given by:

$$\Delta E_i = -(\mathbf{q}_{n+1} - \mathbf{q}_n)^T \mathbf{Q}_{T_{n+\frac{1}{2}}} \quad (12)$$

where  $(\cdot)_{n+\frac{1}{2}}$  denotes evaluation at the midpoint. Note that, in a general nonlinear case,  $\overline{\mathbf{Q}}_{T_{n+\frac{1}{2}}} \neq \mathbf{Q}_{T_{n+\frac{1}{2}}}$  and  $\Delta E_i \neq 0$  can be positive or negative. Note also from (11) and (12) that both numerical schemes are the same and exactly conserve energy ( $\Delta E_i = 0$ ) in the linear case.

Another interesting example is a *conserving algorithm*, which does not introduce artificial energy by means of a specific formulation of the force  $\mathbf{Q}_T^c$ :

$$\Delta E_i = -(\mathbf{q}_{n+1} - \mathbf{q}_n)^T \mathbf{Q}_T^c = 0 \quad (13)$$

Details about the formulation of  $\mathbf{Q}_T^c$  with Cartesian coordinates ( $\mathbf{Q}_T = \mathbf{Q}_\Phi$ ) employing the Lagrange multipliers method and the augmented Lagrange multipliers method can be found in [21] and [19], respectively.

Other expressions similar to (11–13) can be obtained for other integrators, but an exhaustive description falls outside the scope of the work presented here. It is important to remark that the sign of the energy contribution  $\Delta E_i$  may not be constant throughout the simulation, thus increasing or decreasing the total energy, which can in turn affect numerical stability.

The second contribution to the energy variation is  $\Delta E_p$ , associated with the velocity projection described in Sect. 3, and can be obtained solving a minimization problem with a definite positive matrix  $\mathbf{A}$  using a penalty method. This leads to the solution for  $\dot{\mathbf{q}}_{n+1}$  of the linear algebraic equation system (4), given by:

$$\dot{\mathbf{q}}_{n+1} = \mathbf{P}^{-1} \dot{\mathbf{q}}_{n+1}^* \quad \text{with} \quad \mathbf{P} = (\mathbf{1} + \alpha \mathbf{A}^{-1} \Phi_q^T \Phi_q) \quad (14)$$

Introducing the first expression in (14) in the following relation for  $\Delta E_p$ :

$$\Delta E_p = \frac{1}{2}(\dot{\mathbf{q}}_{n+1} + \dot{\mathbf{q}}_{n+1}^*)^T \mathbf{M}(\dot{\mathbf{q}}_{n+1} - \dot{\mathbf{q}}_{n+1}^*)$$

an expression is obtained for the energy variation introduced in the velocity projection:

$$\Delta E_p = \dot{\mathbf{q}}_{n+1}^T \mathbf{D} \dot{\mathbf{q}}_{n+1} \quad \text{with} \quad \mathbf{D} = \frac{1}{2} (\mathbf{I} + \mathbf{P})^T \mathbf{M} (\mathbf{I} - \mathbf{P}) \quad (15)$$

Therefore, the effect of the projection upon the energy depends of the properties of the matrix  $\mathbf{D}$ , which is the matrix associated with the quadratic form  $\Delta E_p$ , and governs the damping behaviour of the projection. If this matrix is negative semidefinite, artificial energy growth is avoided in all cases, and a significant improvement in the stability of the overall numerical scheme would be expected.

In what follows, a detailed analysis of this projection energy balance is performed, which will provide a practical assessment of the suitable choice for projection matrix  $\mathbf{A}$ , so that artificial energy growth is unconditionally avoided.

## 5 Projection Energy Balance

### 5.1 Some Preliminary Results

The point of departure is to perform a quick inspection of the basic properties of the damping matrix  $\mathbf{D}$  based on its definition (15) and some basic linear algebra results. It follows that matrices  $\mathbf{A}^{-1}$  and  $\Phi_q^T \Phi_q$  are symmetric and positive semidefinite. However, matrix  $\mathbf{P}$  defined in (14) is not, in general, positive semidefinite, or even symmetric. As a consequence, the damping matrix  $\mathbf{D}$  given by (15) will not be symmetric, and nothing can be said in general about its definiteness. This means that, following this procedure, it is not possible to bring the sign of the energy balance  $\Delta E_p$  forward at each time step.

Nevertheless, it is possible to get more information about the quadratic form  $\Delta E_p$  as explained in the following proposition:

**Proposition 2.** *The quadratic form  $\Delta E_p$  given by (15) is degenerate, i.e. its kernel  $\mathcal{K}_D$ :*

$$\mathcal{K}_D = \{\mathbf{x} \in \mathbb{R}^n; \mathbf{y}^T \mathbf{D} \mathbf{x} = 0, \forall \mathbf{y} \in \mathbb{R}^n\} \quad (16)$$

*contains other vectors than the zero vector. Specifically, the set  $\mathcal{C}$  of velocity vectors which are compatible with the constraint  $\dot{\Phi}$ :*

$$\mathcal{C} = \{\dot{\mathbf{q}} \in \mathbb{R}^n; \dot{\Phi} = \Phi_q \dot{\mathbf{q}} = \mathbf{0}\}$$

*is a subset of the kernel, thus  $\mathcal{C} \subset \mathcal{K}_D$ .*

A detailed proof of this proposition can be found in [15].

This result was already expected, recalling from Proposition 1 that projected velocities that are compatible come from compatible original velocities, which means that projection leaves them unchanged. Note also that this result does not exclude the possibility that incompatible velocities may be undamped; in other words,  $\mathcal{C}$  may not coincide with  $\mathcal{K}_D$ .

Summarizing, this preliminary inspection reveals that there are few things to say about the damping introduced by the projection with a general definite positive projection matrix  $\mathbf{A}$ , apart from the expected fact that compatible velocities never introduce artificial energy.

The next step is to try to find a set of requirements such that, if satisfied by the projection matrix  $\mathbf{A}$ , they would determine the behaviour of the projection energy balance. If achieved, this result would help in the selection of the projection matrix, which is one of the main goals of this paper.

## 5.2 Conditions for Energy Dissipation

It is a basic linear algebra result that any quadratic form may always be expressed in terms of a symmetric matrix. This means that, in order to analyze the properties of the quadratic form given by (15), it is possible to work just with the symmetric part of matrix  $\mathbf{D}$ . Denoting the symmetric and skew-symmetric parts of the original matrix by superscripts  $s$  and  $h$ , respectively, this result may be expressed as:

$$\mathbf{x}^T \mathbf{D} \mathbf{x} = \mathbf{x}^T \mathbf{D}^s \mathbf{x} + \underbrace{\mathbf{x}^T \mathbf{D}^h \mathbf{x}}_0 \quad \text{for all } \mathbf{x} \in \mathbb{R}^n, \quad (17)$$

where the symmetric matrix  $\mathbf{D}^s$  can be expressed, after some algebraic manipulations using definition (15), as:

$$\mathbf{D}^s = \frac{1}{2} (\mathbf{D} + \mathbf{D}^T) = \frac{1}{2} (\mathbf{M} - \mathbf{P}^T \mathbf{M} \mathbf{P})$$

*Remark 4.* Matrices  $\mathbf{D}$  and  $\mathbf{D}^s$  both have the same definiteness property (definite, semidefinite, etc.), as immediately follows from (17); and their associated quadratic forms have the same kernel  $\mathcal{K}_{\mathbf{D}} = \mathcal{K}_{\mathbf{D}^s}$ .

Matrix  $\mathbf{D}^s$  can be further elaborated and written in terms of the projection matrix  $\mathbf{A}$  and the Jacobian  $\Phi_{\mathbf{q}}$  using (14) for  $\mathbf{P}$ , obtaining:

$$\mathbf{D}^s = -\alpha \left( \mathbf{B}^s + \frac{1}{2} \alpha \mathbf{B}^T \mathbf{M}^{-1} \mathbf{B} \right), \quad (18)$$

$\mathbf{B}$  being a matrix given by:

$$\mathbf{B} = \mathbf{M} \mathbf{A}^{-1} \Phi_{\mathbf{q}}^T \Phi_{\mathbf{q}}, \quad (19)$$

and again denoting the symmetric part of the matrix by the superscript  $s$ . Based on (18), the energy balance of the projection may be expressed as:

$$\begin{aligned} \Delta E_p &= \dot{\mathbf{q}}^T \mathbf{D}^s \dot{\mathbf{q}} = -\alpha \dot{\mathbf{q}}^T \mathbf{B}^s \dot{\mathbf{q}} - \frac{1}{2} \alpha^2 (\mathbf{B} \dot{\mathbf{q}})^T \mathbf{M}^{-1} (\mathbf{B} \dot{\mathbf{q}}) \\ &= \Delta E_{p_1} + \Delta E_{p_2}. \end{aligned} \quad (20)$$



Note that, since the penalty method is employed to solve the minimization problem (2), the projected velocity vector  $\dot{\mathbf{q}}$  may be incompatible (meaning that it may not lie exactly over the velocity constraint manifold  $\dot{\Phi}$ ).

Thus, the projection energy balance is positive or negative depending on the sign of each term  $\Delta E_{p_1}$  and  $\Delta E_{p_2}$  in (20) for an incompatible projected velocity  $\dot{\mathbf{q}} \notin \mathcal{C}$ . (Recall from Proposition 2 that  $\Delta E_p = 0$  for a compatible velocity  $\dot{\mathbf{q}} \in \mathcal{C}$ .)

In [15] a detailed analysis of these terms is carried out, proving the following important proposition:

**Proposition 3.** *If  $\dot{\mathbf{q}}$  is incompatible and matrix  $\mathbf{B}^s = (\mathbf{M}\mathbf{A}^{-1}\Phi_{\mathbf{q}}^T\Phi_{\mathbf{q}})^s$  is positive semidefinite, the projection introduces non-negative energy dissipation,  $\Delta E_p \leq 0$ .*

*What is more, if matrix  $(\mathbf{M}\mathbf{A}^{-1})^s$  is definite, then the projection introduces positive energy dissipation,  $\Delta E_p < 0$ .*

The next proposition justifies the positive performance of projections based on the mass matrix  $\mathbf{M}$ :

**Proposition 4.** *A velocity projection performed with the mass matrix ( $\mathbf{A} = \mathbf{M}$ ) introduces non-negative energy dissipation.*

We are immediately able to prove this based on the previous results, because in this case:

$$\mathbf{B} = \mathbf{M}\mathbf{A}^{-1}\Phi_{\mathbf{q}}^T\Phi_{\mathbf{q}} = \Phi_{\mathbf{q}}^T\Phi_{\mathbf{q}}$$

is always a symmetric and positive semidefinite matrix. Additionally, in this case  $(\mathbf{M}\mathbf{A}^{-1})^s = \mathbf{1}$ , which is a definite matrix, guaranteeing energy dissipation ( $\Delta E_p < 0$ ) for incompatible velocities.  $\square$

Next, two numerical experiments are presented in order to verify the theoretical results outlined in the previous sections.

## 6 Numerical Experiments

### 6.1 Two Particle System

Let us consider a mechanical system composed of two particles with masses  $m_1 = 1$  and  $m_2 = \mu > 0$  moving along a smooth horizontal line, as depicted in Fig. 1. The configuration of the system is defined by the vector of coordinates  $\mathbf{q} = (q_1, q_2)^T$  containing the distances of the particles from a fixed point on the line. In addition, there is a holonomic constraint  $\Phi(\mathbf{q}) = \mathbf{q}^T\mathbf{q} - 1 = q_1^2 + q_2^2 - 1 = 0$ , and as a consequence the system has only one degree of freedom.

The motion starts at  $t = 0$  from position  $\mathbf{q}_0 = (0, 1)^T$  with velocity  $\dot{\mathbf{q}}_0 = (1, 0)^T$ . Taking into account that  $\Phi_{\mathbf{q}} = 2\mathbf{q}$ , it is easy to verify that the constraints at position and velocity levels are satisfied at  $t = 0$ :

$$\Phi_0 = \mathbf{q}_0^T\mathbf{q}_0 - 1 = 0, \quad \dot{\Phi}_0 = \Phi_{\mathbf{q}}^T\dot{\mathbf{q}}_0 = 2(0, 1) \cdot (1, 0)^T = 0$$

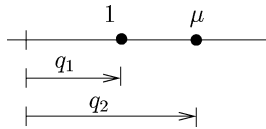
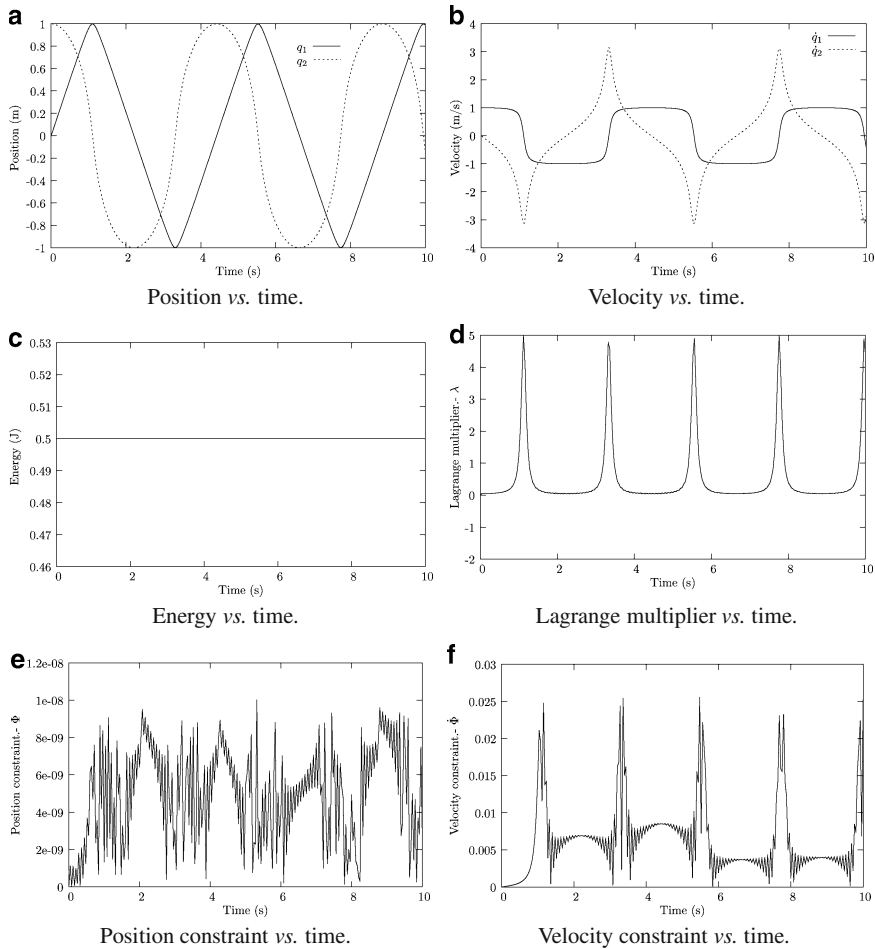


Fig. 1 Two-particle example

Fig. 2 Conserving integration with and without projections,  $\Delta t = 0.1$  s

The motion is integrated up to 10 s with a conserving augmented Lagrangian scheme in position with a penalty of  $10^3$  (see [19] for details of the formulation), such that the constraint at position level is exactly satisfied with exact energy conservation,  $\Delta E_i = 0$ , as expressed in Sect. 4 with expression (13). No projections are performed.

Figures 2a,b show, respectively, the evolution of the position  $\mathbf{q}$  and velocity  $\dot{\mathbf{q}}$  in time for  $\mu = 0.1$  with a constant time step  $\Delta t = 0.04$  s. Figure 2c shows the

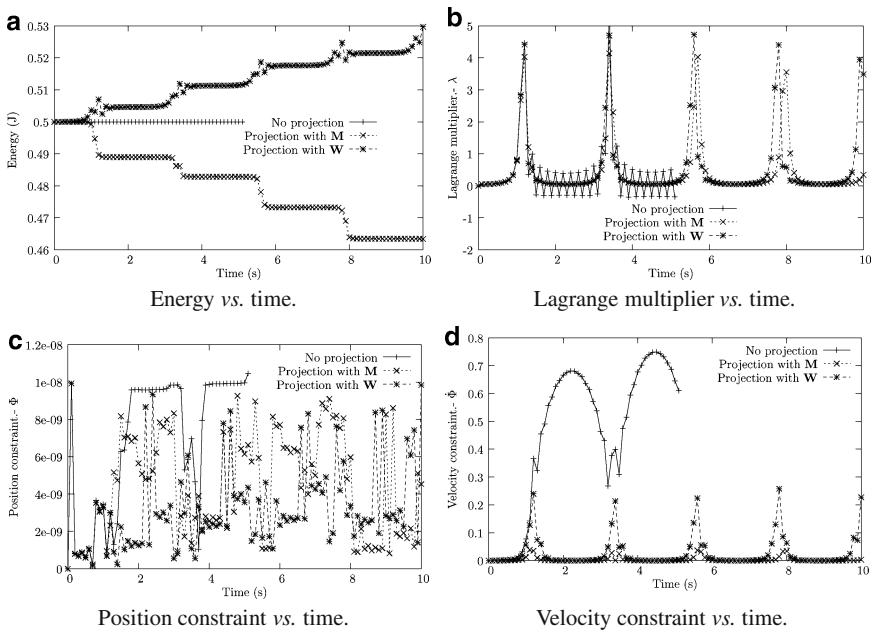
discrete energy, which is exactly constant, as expected, and Fig. 2d shows the Lagrange multiplier, which is related to the constraint force by  $\mathbf{Q}_\Phi = \Phi_q^T \lambda = 2\lambda \mathbf{q}$ .

Figures 2e,f show the constraints at position and velocity levels. It is possible to observe that the position constraint remains small ( $\simeq 10^{-9}$ ), but Fig. 2f shows that the velocity constraint is much larger ( $\simeq 10^{-2}$ ), as expected since no projections are performed. Nevertheless, the energy control performed by the integrator seems capable of handling this undesirable effect, avoiding a noticeable increase in the velocity constraint violation during the integration.

A second set of experiments is performed next, using a larger time integration step  $\Delta t = 0.1$  s. As shown in Fig. 3a, the integration fails to converge at  $t = 5.2$  s when no projections are performed, despite energy being exactly conserved.

Figure 3b shows that the instability at the end of the integration is related to the large oscillations on the Lagrange multiplier. Figure 3c shows that the position constraint is satisfied up to the failure, as expected, but the violation of the velocity constraint shown in Fig. 3d is larger and exhibits a growing trend. Thus, it is reasonable to conclude that the growth of the velocity constraint violation is the ultimate cause of the instability that produces the ultimate failure of the integration.

If a velocity projection with (2) and (4) is performed, the velocity constraint may be significantly reduced and the integration may be carried up to  $t = 10$  s, but the result depends on the projection matrix  $\mathbf{A}$  employed.



**Fig. 3** Conserving integration with and without projections,  $\Delta t = 0.1$  s

Two matrices are tested: the mass matrix  $\mathbf{A} = \mathbf{M} = \begin{pmatrix} 1 & 0 \\ 0 & \mu \end{pmatrix}$  with  $\mu = 0.1$  and a positive definite matrix  $\mathbf{A} = \mathbf{W}$  of the form:

$$\mathbf{W} = \begin{pmatrix} a & 0 \\ 0 & 1 \end{pmatrix} \quad \text{with } a > 0,$$

with  $a = 15$  and a projection penalty parameter  $\alpha = 1$ . This penalty parameter is far too small for practical simulations, but it will serve to more clearly show the performance of the projection technique.

Figures 3c,d show that, as expected, both projection matrices accomplish the task of reducing the velocity constraint, and they also stabilize the integration such that it may be carried up to the end time. But Fig. 3a shows that there are important differences in the behaviour of the energy: while  $\mathbf{M}$  avoids the growth of energy (as predicted in Proposition 4), matrix  $\mathbf{W}$  causes an artificial growth of energy. This behaviour is justified by the properties of matrix  $\mathbf{B}^s$ , which are in this case:

$$\begin{aligned} \mathbf{B} &= \mathbf{M}\mathbf{W}^{-1}\Phi_{\mathbf{q}}^T\Phi_{\mathbf{q}} = 4 \begin{pmatrix} q_1^2/a & q_1q_2/a \\ \mu q_1q_2 & \mu q_2^2 \end{pmatrix} \\ \mathbf{B}^s &= \frac{1}{2}(\mathbf{B} + \mathbf{B}^T) = 2 \begin{pmatrix} 2q_1^2/a & q_1q_2(\mu + 1/a) \\ q_1q_2(\mu + 1/a) & 2\mu q_2^2 \end{pmatrix} \end{aligned}$$

It can be shown that matrix  $\mathbf{B}^s$  is not positive semidefinite, because its determinant is:

$$\det(\mathbf{B}^s) = 4 \left( \frac{4\mu}{a} q_1^2 q_2^2 \right) - q_1^2 q_2^2 \left( \mu + \frac{1}{a} \right)^2 = -\frac{4}{a^2} q_1^2 q_2^2 (1 - a\mu)^2 \leq 0$$

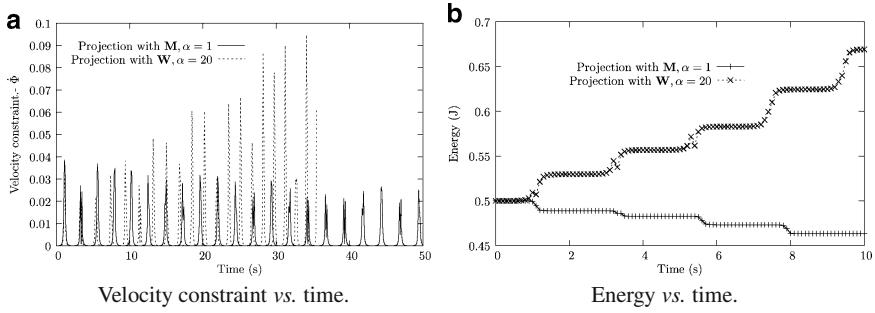
According to Proposition 3, this means that the energy may increase or decrease, which is in fact the behaviour shown in Fig. 3a.

Looking more closely at the projection performed with matrix  $\mathbf{W}$ , a quick inspection of matrix  $\mathbf{M}\mathbf{W}^{-1}$  reveals that it is definite:

$$\mathbf{M}\mathbf{W}^{-1} = \begin{pmatrix} 1/a & 0 \\ 0 & \mu \end{pmatrix},$$

which means that there could be incompatible velocities  $\dot{\mathbf{q}}$  associated with a projection that does not modify the energy ( $\Delta E_p = 0$ ).

Finally, in order to study the effect of the penalty parameter  $\alpha$  in relation to the projection energy balance, a new experiment is performed with the same projection matrix  $\mathbf{W}$  and a larger projection penalty parameter,  $\alpha = 20$ . Figure 4a shows the behaviour of the velocity constraint  $\dot{\Phi}$  in a long simulation, up to 50 s. The projection with  $\mathbf{M}$  and  $\alpha = 1$  retains a small violation of the velocity constraint, producing a stable integration although introducing significant dissipation



**Fig. 4** Conserving integration with projections,  $\Delta t = 0.1$  s



**Fig. 5** Five-bar pendulum, initial configuration

( $E_{t=50}/E_{t=0} \simeq 0.74$ ). On the other hand, the projection with  $W$  and  $\alpha = 20$  controls the violation of the constraint fairly well at the beginning of the simulation; but it fails to keep it small, with integration failure occurring at  $t = 35.5$  s. The large amount of energy introduced by the projection ( $E_{t=35.5}/E_{t=0} \simeq 2.54$ ) shown in Fig. 4b is responsible for the growth of the velocity constraint's violation and the ultimate failure of the computation.

Comparing Figs. 3a,b, it is clear that the artificial energy added to the system by projecting with  $W$  increases significantly when the projection penalty parameter  $\alpha$  increases. While the energy at the end of the computation for  $\alpha = 1$  is  $E_{t=10} \simeq 0.53$  J (as shown in Fig. 3a), the energy for  $\alpha = 20$  shown in Fig. 4b is  $E_{t=10} \simeq 0.67$  J.

## 6.2 Five-Bar Pendulum

This example is analysed as a benchmark problem in several references [7, 14]. The system is composed by five prismatic bars with unitary length, section  $0.1 \times 0.1$  m<sup>2</sup> and unitary masses, linked by revolute joints, being one of them fixed. The system is released from rest under the action of gravity  $g = 9.81$  m/s<sup>2</sup> from the configuration shown in Fig. 5.

The configuration is defined by a set of dependent coordinates collected in vector  $\mathbf{q} \in \mathbb{R}^{60}$ , which are inertial cartesian coordinates of four non-coplanary points at each bar. These coordinates are related by a set of 55 constraint equations collected in vector  $\Phi$ , where  $6 \times 5 = 30$  equations correspond to the constant-distance constraints among the four points of each bar. There are 25 more equations related to the five revolute joints, with five constraints for each joint. Based on these considerations, the system has a total number of five degrees of freedom.

The motion is integrated during 10 s with the trapezoidal rule, employing an augmented Lagrangian method for the enforcement of the position constraint and using different time steps and velocity projections. The matrices employed for the projections are the mass matrix  $\mathbf{M}$  and the identity matrix  $\mathbf{I}$ . As pointed out in Sect. 3, from a purely mathematical point of view both matrices are suitable for projection, as long as both are symmetric and positive definite. However, it will be shown next that they perform very differently in terms of stability, justified by the behaviour of the artificial energy that they introduce.

A first numerical experiment is performed with a time step  $\Delta t = 0.005$  s. Figure 6 shows that the integration fails at  $t \simeq 2$  s if no projections are performed, showing a dramatical increase of energy which is clearly related to an abrupt increase of the velocity constraint. If a projection is performed with the mass matrix  $\mathbf{M}$  and a projection penalty parameter  $\alpha = 10^7$ , the velocity constraint remains small during all the computation and the energy remains very close to the theoretical constant value.

Note that the projection with  $\mathbf{M}$  introduces artificial dissipation (which is always positive, as justified in Proposition 4), but the trapezoidal rule itself introduces some artificial energy too, which may be positive or negative, as remarked in Sect. 4. This explains the behavior of the total energy observed in Fig. 6a, which is not monotonically decreasing as in the case of the conserving integrator employed in the two-particle example presented in Sect. 6.1.

On the other hand, Fig. 6b shows that the projection performed with the identity matrix succeeds in enforcing the velocity constraint. However, Fig. 6a reveals that it

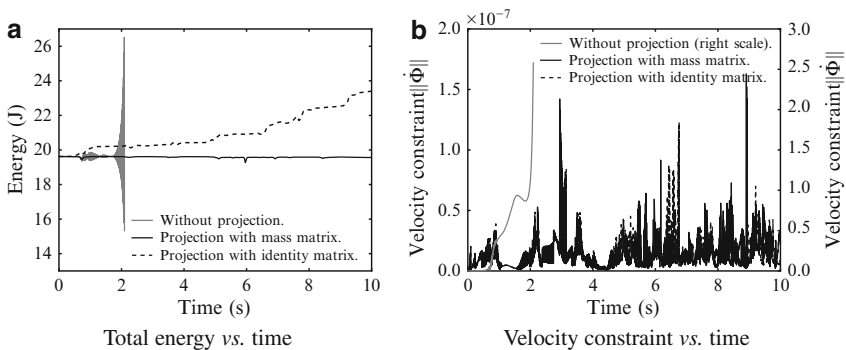


Fig. 6 Five-bar pendulum, trapezoidal rule with  $\Delta t = 0.005$  s

introduces a significant amount of artificial positive energy, which is unrealistic and may eventually produce a failure of the computation. This behaviour is fully justified by Proposition 3, since  $\mathbf{B}^s = (\mathbf{M}\Phi_q^T\Phi_q)^s$  is not, in general, a positive semidefinite matrix.

The results of a second experiment performed with a larger time step,  $\Delta t = 0.01$  s, are shown in Fig. 7. These results are very similar to those presented in Fig. 6 for a smaller time step. Figure 7b shows that both projections succeed in enforcing the velocity constraint, but there is a significant difference in the behaviour of the energy, as shown in Fig. 7a. It is apparent that the increment of the time step amplifies the growth of the artificial positive energy introduced by the projection with the identity matrix, deteriorating the overall performance of the computation.

A third set of experiments is performed in order to explore the effect of the projection penalty for a fixed time step. Figure 8 shows that the overall performance of the mass matrix projection is very good. Figure 8a reveals that the energy is not very affected by the projection penalty value, while Fig. 8b shows that the larger projection penalty improves the satisfaction of the velocity constraint, as expected.

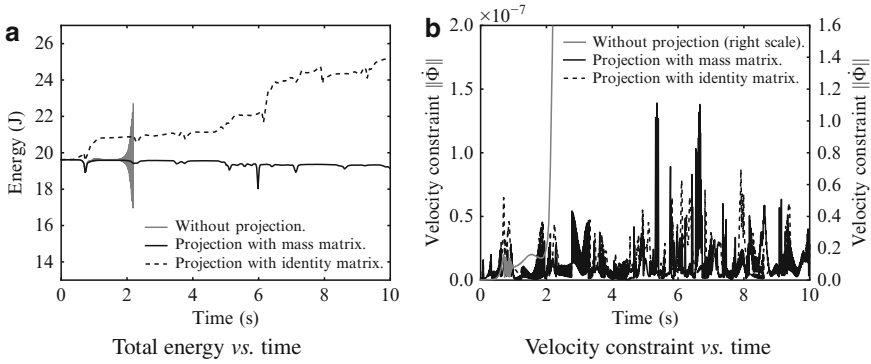


Fig. 7 Five-bar pendulum, trapezoidal rule with  $\Delta t = 0.01$  s

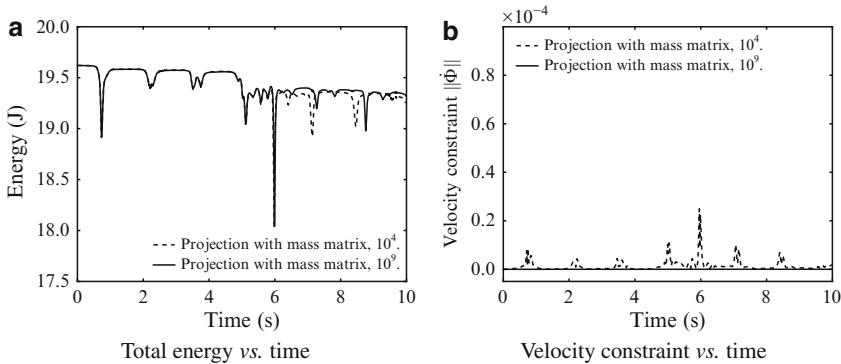
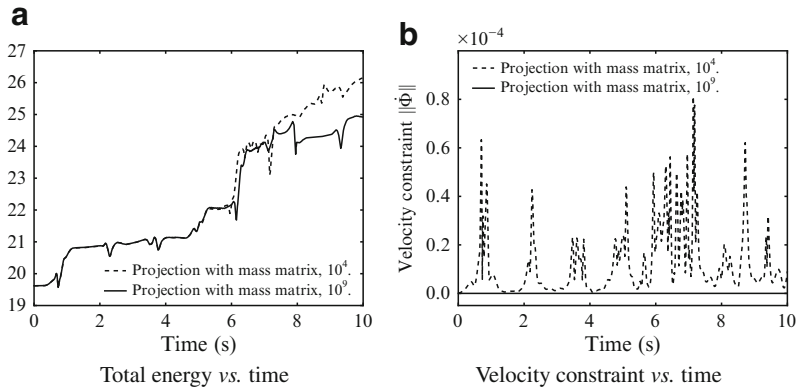


Fig. 8 Five-bar pendulum, trapezoidal rule with  $\Delta t = 0.01$  s, projection with mass matrix



**Fig. 9** Five-bar pendulum, trapezoidal rule with  $\Delta t = 0.01$  s, projection with identity matrix

Figure 9 shows the results obtained with the identity projection matrix. Figure 9b shows that the larger projection penalty improves the satisfaction of the velocity constraint, as expected. But note in Fig. 9a how this effect does not help to significantly reduce the growth of the artificial positive energy introduced by the projection.

## 7 Conclusions

The main conclusions that may be drawn from the developments presented in this work are:

- A velocity projection, solving a minimization problem based on a positive definite matrix and using a penalty method, succeeds in maintaining the numerical solution of the index-3 DAE system close to the velocity constraint manifold  $\dot{\Phi} = 0$ . This projection has a stabilization effect that has been reported in the literature and tested with a simple example in this paper.
- The velocity projection may introduce some artificial energy into the system. If this energy is negative (dissipation) the stabilization effect of the projections is enhanced, making it possible to adopt larger integration time steps or allowing longer term computations.

On the other hand, a positive energy spoils the stabilization effect introduced by the projections, resulting in an unrealistic motion and eventually a failure of the computations.

- The consequence of the previous statement is that not all positive definite matrices are suited to performing a useful projection. Some positive artificial energy may be introduced into the system, compromising the stability of the numerical scheme. The numerical experiment presented in this paper, despite its simplicity, shows this effect very clearly.



- For a system with a mass matrix  $\mathbf{M}$ , subject to a holonomic constraint function vector  $\Phi(\mathbf{q})$ , a velocity projection does not increase the energy of the system if the symmetric part of the matrix  $\mathbf{M}\mathbf{A}^{-1}\Phi_{\mathbf{q}}^T\Phi_{\mathbf{q}}$  is positive semidefinite,  $\mathbf{A}$  being the projection matrix.

This property provides a practical criterion for the selection of a projection matrix, which is an important issue that is not explicitly discussed in the literature in this field.

- Additionally, if the symmetric part of the matrix  $\mathbf{M}\mathbf{A}^{-1}$  is definite, the projection of incompatible velocities always introduces energy dissipation.
- Finally, the energy balance of the velocity projection provides an alternative justification for the positive performance of the mass-orthogonal projection reported in the literature.

**Acknowledgements** The authors wish to acknowledge the financial support of the Spanish Ministry of Science and Innovation as part of project DPI-2006-15613-C03-02 under the name “Modelización numérica eficiente de grandes sistemas flexibles con aplicaciones de impacto”.

## References

1. Alishenas T, Ólafsson Ö (1994) Modeling and velocity stabilization of constrained mechanical systems. *BIT Numer Math* 34:455–483
2. Arnold M, Bruls O (2007) Convergence of the generalized-alpha scheme for constrained mechanical systems. *Multibody Syst Dyn* 18:185–202
3. Ascher UM (1997) Stabilization of invariants of discretized differential systems. *Numer Algorithm* 14:1–23
4. Ascher UM, Chin H, Petzold LR, Reich S (1995) Stabilization of constrained mechanical systems with DAEs and invariant manifolds. *J Mech Struct Mach* 23:135–158
5. Ascher UM, Petzold LR (1998) Computer methods for ordinary differential equations and differential-algebraic equations. SIAM, Philadelphia, PA
6. Bauchau OA, Laulusa A (2008) Review of contemporary approaches for constraint enforcement in multibody systems. *J Comput Nonlin Dynam (ASME)* 3:1–8
7. Bayo E, Ledesma R (1996) Augmented Lagrangian and mass-orthogonal projection methods for constrained multibody dynamics. *Nonlinear Dynam* 9:113–130
8. Brenan KE, Campbell SL, Petzold LR (1996) Numerical solution of initial-value problems in differential-algebraic equations, SIAM, Philadelphia, PA, <http://www.ec-securehost.com/SIAM/CL14.html>
9. Cuadrado J, Cardenal J, Bayo E (1997) Modeling and solution methods for efficient real-time simulation of multibody dynamics. *Multibody Syst Dyn* 1:259–280
10. Cuadrado J, Cardenal J, Morer P, Bayo E (2000) Intelligent simulation of multibody dynamics: space-state and descriptor methods in sequential and parallel computing environments. *Multibody Syst Dyn* 4:55–73
11. Cuadrado J, Dopico D, Naya M, González M (2004) Penalty, semi-recursive and hybrid methods for MBS real-time dynamics in the context of structural integrators. *Multibody Syst Dyn* 12:117–132
12. Eich E (1993) Convergence results for a coordinate projection method applied to mechanical systems with algebraic constraints. *SIAM J Numer Anal* 30(5):1467–1482
13. Eich E, Führer C, Leimkhuler B, Reich S (1990) Stabilization and projection methods for multibody dynamics – Helsinki Institute of Technology. Technical Report A281

14. García de Jalón J, Bayo E (1994) Kinematic and dynamic simulation of multibody systems. The real time challenge. Springer, New York
15. García Orden JC (2009) Energy considerations for the stabilization of constrained mechanical systems with velocity projection. *Nonlinear Dynam.* doi:10.1007/s11071-009-9579-8
16. García Orden JC, Dopico Dopico D (2007) On the stabilizing properties of energy-momentum integrators and coordinate projections for constrained mechanical systems. In: *Multibody Dynamics. Computational Methods in Applied Sciences*. Springer, The Netherlands, pp 49–67
17. García Orden JC, Goicolea JM (2000) Conserving properties in constrained dynamics of flexible multibody systems. *Multibody Syst Dyn* 4:225–244
18. García Orden JC, Goicolea JM (2005) Robust analysis of flexible multibody systems and joint clearances in an energy conserving framework. In: *Advances in Computational Multibody Dynamics. Computational Methods in Applied*. Springer, The Netherlands, pp 205–237
19. García Orden JC, Ortega R (2006) A conservative augmented Lagrangian algorithm for the dynamics of constrained mechanical systems. *Mech Base Des Struct Mach* 34(4):449–468
20. Goicolea JM, García Orden JC (2002) Quadratic and higher-order constraints in energy-conserving formulations in flexible multibody systems. *Multibody Syst Dyn* 7:3–29
21. González O (1999) Mechanical systems subjected to holonomic constraints: Differential-algebraic formulations and conservative integration. *Physica D* 132:165–174
22. Hairer E, Wanner G (1991) Solving ordinary differential equations II, stiff and differential-algebraic problems. Springer, Berlin
23. Hughes T (1987) The finite element method. Prentice-Hall, Englewood Cliffs, NJ
24. Ortiz M (1986) A note on energy conservation and stability of nonlinear time-stepping algorithms. *Comput Struct* 24(1):167–168
25. Lubich C (1991) Extrapolation integrators for constrained multibody systems. *Impact Comput Sci Eng* 3:213–234
26. Stuart A, Humphries A (1996) Dynamical systems and numerical analysis. Cambridge University Press, Cambridge

# A General Purpose Algorithm for Optimal Trajectory Planning of Closed Loop Multibody Systems

Makoto Iwamura, Peter Eberhard, Werner Schiehlen, and Robert Seifried

**Abstract** This paper discusses the optimal trajectory planning problem of multi-body systems. The aim of this study is to develop a general purpose optimal trajectory planning algorithm to be applied to arbitrary multibody systems. Multi-body systems may be divided into two groups, i.e. open loop systems and closed loop systems [8]. In [11] an optimal trajectory planning algorithm for open loop systems was presented. In this paper, optimal trajectory planning algorithms for closed loop systems are proposed by extending the algorithm for open loop systems. Two types of methods are presented based on the dynamic analysis by computational algorithms for closed loop systems. The first method uses generalized coordinate partitioning and embedding techniques. The second method is based on an augmented formulation with Lagrange multipliers. The first method is easily applicable to non-redundant actuation systems, while the second method considers redundant actuation. The validity of these methods for optimal trajectory planning is confirmed by computational results and their features are compared.

## 1 Introduction

Optimal trajectory planning of multibody systems deals with the problem to find a trajectory between a specified initial and final state that minimizes a given cost function. This topic has been studied by many researchers as an important problem especially in the fields of robotics and space engineering, see e.g. [1, 3, 4, 7, 9, 10, 13, 20]. However, most of the previous papers discuss the problem of

---

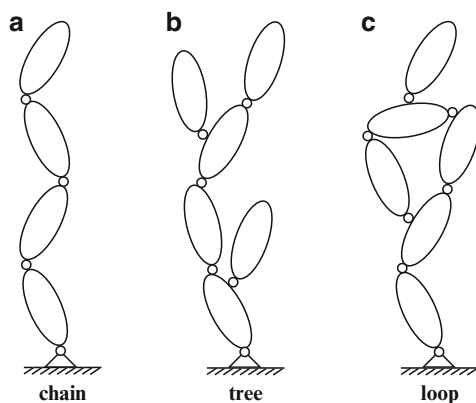
M. Iwamura (✉)

Department of Mechanical Engineering, Fukuoka University, Fukuoka, Japan  
e-mail: [iwamura@cis.fukuoka-u.ac.jp](mailto:iwamura@cis.fukuoka-u.ac.jp)

P. Eberhard, W. Schiehlen, and R. Seifried

Institute of Engineering and Computational Mechanics, University of Stuttgart, Stuttgart, Germany  
e-mail: [eberhard@itm.uni-stuttgart.de](mailto:eberhard@itm.uni-stuttgart.de); [schiehlen@itm.uni-stuttgart.de](mailto:schiehlen@itm.uni-stuttgart.de);  
[seifried@itm.uni-stuttgart.de](mailto:seifried@itm.uni-stuttgart.de)

**Fig. 1** Topology of multibody systems,  
 (a) open chain system,  
 (b) tree-structured system,  
 (c) closed loop system



a specific multibody system, such as a manipulator, space robot, biped robot, etc., while a general purpose optimal trajectory planning algorithm which can be applied to arbitrary multibody systems is missing. Hence, this study considers the development of a general purpose optimal trajectory planning algorithm for multibody systems by extending the dynamics formalisms for multibody systems. The goal of this study is to develop a software that can conduct optimal trajectory calculation easily by inputting only the kinematical and dynamical parameters, and the cost function.

Multibody systems may be divided into two groups, i.e. open loop systems and closed loop systems, see Fig. 1 and [8]. In [11], an optimal trajectory planning algorithm for open loop systems, as shown in Fig. 1a,b, was presented. In this study, a general purpose optimal trajectory planning algorithm is established by extending the algorithm to closed loop systems, as shown in Fig. 1c.

For solving the optimal trajectory planning problem, two different approaches are commonly used, i.e. the exact method based on the minimum principle, see e.g. [1,3,10,13] and the approximate method using a Ritz method, see e.g. [4,7,9,20]. In this study, the former approach is considered. The solution of the optimal trajectory planning problems based on the minimum principle requires sensitivity analysis of the dynamic equations, i.e. the partial derivatives of the state equation with respect to the states and the control variables have to be calculated.

In the modeling of closed loop systems, a method that transforms the closed loop systems to open loop tree-structured systems by cutting the loop virtually and adding algebraic loop closing conditions [14,15,21] is well known. In this paper, two types of optimal trajectory planning algorithm that use this concept of virtual open loop tree-structured systems are proposed. Firstly, an optimal trajectory planning algorithm is formulated based on generalized coordinate partitioning and embedding techniques [8] that derives the minimum number of differential equations of motion by eliminating the dependent coordinates. This method is exact in the sense that it satisfies the loop closing conditions completely, but it is difficult to apply to redundant actuation systems, characterized by more actuators than degrees of

freedom. Let us denote the number of actuators included in the closed loop system by  $n_a$ . Generally, closed loop systems may be designed as  $n_a \geq n$ . If  $n_a > n$  the system is called a redundant actuation system [15]. Hence, secondary, another optimal trajectory planning algorithm based on an augmented formulation [8] is proposed. This method can also be applied to redundant actuation systems but it might yield small violations of the loop closing conditions.

## 2 Optimal Trajectory Planning Problem of Multibody Systems

In this section the basics of the two proposed methods are presented.

### 2.1 Problem Formulation 1: Minimal Form of Equation of Motion

The equations of motion of multibody systems can be expressed by using the generalized coordinates  $\mathbf{q} \in R^n$  and the generalized control forces  $\boldsymbol{\tau} \in R^n$  as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau}, \quad (1)$$

where  $\mathbf{M}(\mathbf{q}) \in R^{n \times n}$  is the inertia matrix,  $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) \in R^n$  is the vector of centrifugal, and Coriolis forces, gravity, and any applied forces other than  $\boldsymbol{\tau}$ . Equation (1) can be rewritten

$$\frac{d\mathbf{q}}{dt} = \dot{\mathbf{q}}, \quad (2)$$

$$\frac{d\dot{\mathbf{q}}}{dt} = \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}), \quad (3)$$

where  $\mathbf{h} : \mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau} \mapsto \ddot{\mathbf{q}}$  is the function defining the acceleration

$$\ddot{\mathbf{q}} = \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}) = \mathbf{M}^{-1}(\mathbf{q})\{\boldsymbol{\tau} - \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}})\}. \quad (4)$$

We define the state vector as  $\mathbf{x} = [\mathbf{q}^T \ \dot{\mathbf{q}}^T]^T \in R^{2n}$  and the input vector as  $\mathbf{u} = \boldsymbol{\tau}$ . Then, the state equations can be expressed in the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}). \quad (5)$$

We assume that the initial and final states are given as  $\mathbf{x}(0) = \mathbf{x}_0$  for  $t = 0$  and  $\mathbf{x}(t_f) = \mathbf{x}_f$  for  $t = t_f$ . The cost function is chosen as

$$J = \int_0^{t_f} F(\mathbf{x}, \mathbf{u})dt. \quad (6)$$

The problem considered in this paper can be formulated as an optimal control problem as follows: Find  $\mathbf{u}(t)$  by minimizing the cost function (6) subject to the initial and final conditions for the system (5).

## 2.2 Problem Formulation 2: Augmented Equations

For closed loop systems, (1) and (5) may not be available directly. Therefore, we transform a closed loop system into an open loop tree-structured system by cutting the loop at one of its joints, see Fig. 2. The equations of motion of virtual open loop tree-structured systems read as

$$\bar{\mathbf{M}}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \bar{\mathbf{c}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \boldsymbol{\tau}_0, \quad (7)$$

where  $\boldsymbol{\theta} \in R^{n_0}$  is the vector of joint variables,  $\boldsymbol{\tau}_0 \in R^{n_0}$  is the vector of driving torques/forces,  $\bar{\mathbf{M}}(\boldsymbol{\theta}) \in R^{n_0 \times n_0}$  is the inertia matrix,  $\bar{\mathbf{c}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \in R^{n_0}$  is the vector of centrifugal, Coriolis and applied forces of the virtual open loop tree-structured system. The equations of motion for the original closed loop system can be written by adding the loop closing conditions  $\Phi(\boldsymbol{\theta}) = \mathbf{0} \in R^m$  between the cut branches to (7) as

$$\bar{\mathbf{M}}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \bar{\mathbf{c}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \Phi_{\boldsymbol{\theta}}^T \boldsymbol{\lambda} = \boldsymbol{\tau}_0, \quad (8)$$

$$\Phi(\boldsymbol{\theta}) = \mathbf{0}, \quad (9)$$

where  $\Phi_{\boldsymbol{\theta}} = \partial\Phi/\partial\boldsymbol{\theta} \in R^{m \times n_0}$  and  $\boldsymbol{\lambda} \in R^m$  is the Lagrange multipliers vector representing the constraint reaction forces originating from the cut joints. The closed loop system has  $n = n_0 - m$  degrees of freedom. Equations (8) and (9) can be rewritten as

$$\frac{d\boldsymbol{\theta}}{dt} = \dot{\boldsymbol{\theta}}, \quad (10)$$

$$\frac{d\dot{\boldsymbol{\theta}}}{dt} = \bar{\mathbf{h}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \boldsymbol{\tau}_0), \quad (11)$$

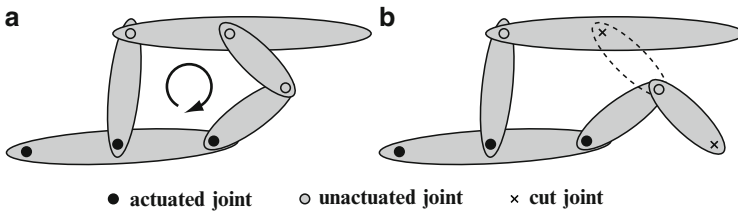


Fig. 2 (a) A closed loop system, and (b) its transformation to a virtual tree-structured system

where  $\bar{\mathbf{h}} : \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \boldsymbol{\tau}_0 \mapsto \ddot{\boldsymbol{\theta}}$  is a function which is derived in Sect. 4. We define the state vector as  $\bar{\mathbf{x}} = [\boldsymbol{\theta}^T \dot{\boldsymbol{\theta}}^T]^T$  and the input vector as  $\bar{\mathbf{u}} = \boldsymbol{\tau}_0$ . Then, the state equations can be expressed in the form

$$\dot{\bar{\mathbf{x}}} = \bar{\mathbf{f}}(\bar{\mathbf{x}}, \bar{\mathbf{u}}). \quad (12)$$

We assume that the initial and final states are given as  $\bar{\mathbf{x}}(0) = \bar{\mathbf{x}}_0$  for  $t = 0$  and  $\bar{\mathbf{x}}(t_f) = \bar{\mathbf{x}}_f$  for  $t = t_f$ . The cost function is

$$J = \int_0^{t_f} \bar{F}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) dt. \quad (13)$$

Then, the problem considered in this paper can also be stated as follows: Find  $\bar{\mathbf{u}}(t)$  by minimizing the cost function (13) subject to the initial and final conditions for the system (12).

### 2.3 Solution Procedure for Optimal Trajectory Planning Problems

The minimum principle, see [18], is well-known as a mathematical tool for solving optimal control problems. In the following, we consider the case of formulation 1, but the solution procedure for the case of formulation 2 is the same. Let us introduce a vector  $\boldsymbol{\psi} \in R^{2n}$  called adjoint vector. Then, the Hamiltonian is defined by using the adjoint vector as

$$H(\boldsymbol{\psi}, \mathbf{x}, \mathbf{u}) = F(\mathbf{x}, \mathbf{u}) + \boldsymbol{\psi}^T \mathbf{f}(\mathbf{x}, \mathbf{u}). \quad (14)$$

The behavior of  $\mathbf{x}(t)$  and  $\boldsymbol{\psi}(t)$  is determined by the canonical equations of Hamilton

$$\dot{\mathbf{x}} = \frac{\partial H}{\partial \boldsymbol{\psi}} = \mathbf{f}, \quad (15)$$

$$\dot{\boldsymbol{\psi}} = -\frac{\partial H}{\partial \mathbf{x}} = -\frac{\partial F}{\partial \mathbf{x}} - \boldsymbol{\psi}^T \frac{\partial \mathbf{f}}{\partial \mathbf{x}}, \quad (16)$$

where (15) is equivalent to the state equation (5), and (16) is called the adjoint equation. The minimum principle [18] states that the necessary condition for  $\mathbf{u}(t)$  to be optimal is that there exists a nonzero vector  $\boldsymbol{\psi}$  that satisfies (15), (16), and the Hamiltonian (14) is minimized for all times  $t$ .

The solution procedure based on the minimum principle is as follows. At first, the form of optimal control  $\mathbf{u}^{opt}$  that minimizes the Hamiltonian (14) is derived as a function of  $\mathbf{x}$  and  $\boldsymbol{\psi}$ . Then,  $\mathbf{u}^{opt}(\mathbf{x}, \boldsymbol{\psi})$  is substituted into (15) and (16). After that the differential equations (15) and (16) are solved for  $\mathbf{x}$ ,  $\boldsymbol{\psi}$  under the two-point boundary conditions. Finally, the optimal control can be obtained by substituting  $\mathbf{x}$ ,  $\boldsymbol{\psi}$  into  $\mathbf{u}^{opt}(\mathbf{x}, \boldsymbol{\psi})$ . Since the two-point boundary value problem is difficult to solve

analytically, a numerical algorithm is usually used. Moreover, in the case that the analytical form of the optimal control  $\mathbf{u}^{opt}$  can not be obtained, one should solve the two-point boundary value problem iteratively by improving  $\mathbf{u}$  as  $\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + \delta\mathbf{u}^{(i)}$  so that  $H$  decreases, based on the information from the gradient

$$\frac{\partial H}{\partial \mathbf{u}} = \frac{\partial F}{\partial \mathbf{u}} + \boldsymbol{\psi}^T \frac{\partial \mathbf{f}}{\partial \mathbf{u}}. \quad (17)$$

The simplest method is to choose  $\delta\mathbf{u} = -\partial H/\partial \mathbf{u}$  but there are many other more sophisticated algorithms to compute  $\delta\mathbf{u}$ , see e.g. [5, 12, 19].

A typical computational procedure for optimal trajectory planning problems is shown in Fig. 3. The solution of the optimal trajectory planning problem requires a forward dynamics computation including the computation of the state equations  $\mathbf{f}$  and a sensitivity analysis of dynamic equations, i.e. computing the partial derivatives of the state equations with respect to the state and control variables  $\partial\mathbf{f}/\partial\mathbf{x}$ ,  $\partial\mathbf{f}/\partial\mathbf{u}$ . One can compute  $\mathbf{f}$  over time by using existing forward dynamics algorithms. On the other hand, a general method for computing these derivatives is not yet available. For the case of systems with only few degrees of freedom, it is possible to derive the closed-form equations of  $\partial\mathbf{f}/\partial\mathbf{x}$ ,  $\partial\mathbf{f}/\partial\mathbf{u}$ . However, it becomes difficult to obtain

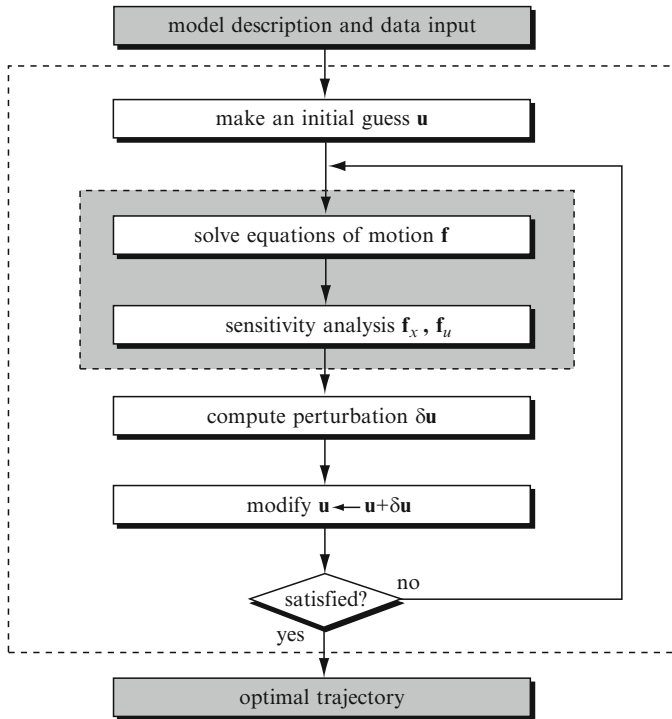


Fig. 3 Computational procedure for optimal trajectory planning problems



the closed-form equations analytically if the degrees of freedom or its complexity increases. Although finite difference methods can be used to obtain approximate values of the derivatives, its inaccuracy can affect the convergence of the optimization process. In [11], a general and exact method to compute  $\partial \mathbf{f}/\partial \mathbf{x}$ ,  $\partial \mathbf{f}/\partial \mathbf{u}$  for open loop systems was presented. In this paper, methods to compute  $\partial \mathbf{f}/\partial \mathbf{x}$ ,  $\partial \mathbf{f}/\partial \mathbf{u}$  (or  $\partial \bar{\mathbf{f}}/\partial \bar{\mathbf{x}}$ ,  $\partial \bar{\mathbf{f}}/\partial \bar{\mathbf{u}}$ ) for closed loop systems are developed by extending the algorithm for open loop systems. The procedure for computing the optimal control  $\mathbf{u}$  (or  $\bar{\mathbf{u}}$ ) after obtaining  $\mathbf{f}$  (or  $\bar{\mathbf{f}}$ ) and the derivatives  $\partial \mathbf{f}/\partial \mathbf{x}$ ,  $\partial \mathbf{f}/\partial \mathbf{u}$  (or  $\partial \bar{\mathbf{f}}/\partial \bar{\mathbf{x}}$ ,  $\partial \bar{\mathbf{f}}/\partial \bar{\mathbf{u}}$ ) is similar to the procedure presented in [11].

### 3 Optimal Trajectory Planning Algorithm Using Coordinate Partitioning and Embedding Techniques

In this section only non-redundant actuation systems, i.e.  $n_a = n$ , are considered and an optimal control algorithm based on problem formulation 1 is developed. Since for closed loop systems it is difficult to compute  $\mathbf{f}$ ,  $\partial \mathbf{f}/\partial \mathbf{x}$ ,  $\partial \mathbf{f}/\partial \mathbf{u}$  directly, we compute these quantities by using the concept of the virtual open loop tree-structured system, i.e. (8), (9) and use a coordinate partitioning technique to derive the equation of motion with minimal order of form (1).

Differentiating (9) with respect to time leads to

$$\dot{\Phi}(\theta) = \Phi_\theta \dot{\theta} = [\Phi_q \ \Phi_p] \begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{p}} \end{bmatrix} = \mathbf{0}, \quad (18)$$

where the vector of joint variables of the virtual open loop tree-structured system  $\theta \in R^{n_0}$  is partitioned by the independent coordinates  $\mathbf{q} \in R^n$  and the dependent coordinates  $\mathbf{p} \in R^m$ , and  $\Phi_\theta \in R^{m \times n_0}$  is also divided into  $\Phi_q \in R^{m \times n}$  and  $\Phi_p \in R^{m \times m}$  corresponding to the coordinate partitioning. If there are  $m$  independent constraints, we can assume that  $\Phi_p$  is nonsingular without loss of generality because  $\Phi_\theta$  has  $m$  independent columns. Thus from (18), one obtains an expression for the velocity of the dependent coordinates as

$$\dot{\mathbf{p}} = -\Phi_p^{-1} \Phi_q \dot{\mathbf{q}}. \quad (19)$$

If the independent coordinates  $\mathbf{q}$  are chosen as a subset of the coordinates  $\theta$  of the virtual tree-structured system, then  $\dot{\theta}$  can be written in terms of  $\dot{\mathbf{q}}$  as

$$\dot{\theta} = \mathbf{W} \dot{\mathbf{q}}, \quad (20)$$

where

$$\mathbf{W} = \begin{bmatrix} \mathbf{E} \\ -\Phi_p^{-1} \Phi_q \end{bmatrix}, \quad (21)$$

is a Jacobian matrix and  $\mathbf{E}$  is an identity matrix. We can compute  $\Phi_p$  and  $\Phi_q$  by using the Jacobian computational method for serial link manipulators, see e.g. Orin's method [17].

Differentiating (20) with respect to time yields

$$\ddot{\theta} = \mathbf{W}\ddot{\mathbf{q}} + (\mathbf{W}\dot{\mathbf{q}})_q\dot{\mathbf{q}} = \mathbf{W}\ddot{\mathbf{q}} + \mathbf{G}\dot{\mathbf{q}}^2, \quad (22)$$

where

$$\mathbf{G} = \mathbf{W}_q = \begin{bmatrix} \mathbf{O} \\ -\Phi_p^{-1}\Phi_{pq}\Phi_p^{-1}\Phi_q - \Phi_p^{-1}\Phi_{qq} \end{bmatrix} \quad (23)$$

is a Hessian matrix and  $\mathbf{O}$  is a zero matrix. We can compute  $\Phi_{pq}$  and  $\Phi_{qq}$  by using the Hessian computational method for serial link manipulators, see e.g. Nakamura's method [16].

Substituting (22) into (8) leads to

$$\overline{\mathbf{M}}(\mathbf{W}\ddot{\mathbf{q}} + \mathbf{G}\dot{\mathbf{q}}^2) + \bar{\mathbf{c}} + \Phi_\theta^T \lambda = \tau_0. \quad (24)$$

Premultiplying this equation by  $\mathbf{W}^T$  becomes

$$\mathbf{W}^T \overline{\mathbf{M}} \mathbf{W} \ddot{\mathbf{q}} + \mathbf{W}^T (\overline{\mathbf{M}} \mathbf{G} \dot{\mathbf{q}}^2 + \bar{\mathbf{c}}) = \mathbf{W}^T \tau_0, \quad (25)$$

where the principle of virtual work

$$\mathbf{W}^T \Phi_\theta^T = \begin{bmatrix} \mathbf{E} - (\Phi_p^{-1}\Phi_q)^T \end{bmatrix} \begin{bmatrix} \Phi_q^T \\ \Phi_p^T \end{bmatrix} = \mathbf{0} \quad (26)$$

is used. By comparing (25) with (1), we can obtain the following relationships

$$\mathbf{M}(\mathbf{q}) = \mathbf{W}^T \overline{\mathbf{M}} \mathbf{W}, \quad (27)$$

$$\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{W}^T (\overline{\mathbf{M}} \mathbf{G} \dot{\mathbf{q}}^2 + \bar{\mathbf{c}}), \quad (28)$$

$$\boldsymbol{\tau} = \mathbf{W}^T \tau_0. \quad (29)$$

These equations allow to compute the quantities of the original closed loop system  $\mathbf{M}, \mathbf{c}, \boldsymbol{\tau}$  from the quantities of the virtual open loop tree-structured system  $\overline{\mathbf{M}}, \bar{\mathbf{c}}, \tau_0$  by using  $\mathbf{W}, \mathbf{G}$ .

Next, we consider the partial derivative of  $\mathbf{f}$  with respect to  $\mathbf{x}$ . From (2) and (3),  $\partial \mathbf{f} / \partial \mathbf{x}$  can be expressed as

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}} & \frac{\partial \dot{\mathbf{q}}}{\partial \dot{\mathbf{q}}} \\ \frac{\partial \ddot{\mathbf{q}}}{\partial \mathbf{q}} & \frac{\partial \ddot{\mathbf{q}}}{\partial \dot{\mathbf{q}}} \end{bmatrix} = \begin{bmatrix} \mathbf{O} & \mathbf{E} \\ \frac{\partial \ddot{\mathbf{q}}}{\partial \mathbf{q}} & \frac{\partial \ddot{\mathbf{q}}}{\partial \dot{\mathbf{q}}} \end{bmatrix}. \quad (30)$$

In order to compute  $\partial \mathbf{f} / \partial \mathbf{x}$  the quantities  $\partial \ddot{\mathbf{q}} / \partial \mathbf{q}$  and  $\partial \ddot{\mathbf{q}} / \partial \dot{\mathbf{q}}$  must be computed. In [11], the following equations are derived for open loop systems

$$\frac{\partial \ddot{\mathbf{q}}}{\partial \mathbf{q}} = -\mathbf{M}^{-1} \frac{\partial \boldsymbol{\tau}}{\partial \mathbf{q}}, \quad (31)$$

$$\frac{\partial \ddot{\mathbf{q}}}{\partial \dot{\mathbf{q}}} = -\mathbf{M}^{-1} \frac{\partial \boldsymbol{\tau}}{\partial \dot{\mathbf{q}}}. \quad (32)$$

This relationship does hold even in the case of closed loop systems since the equations of motion (1) have the same form as the one of open loop systems. Hence, the problem reduces to the computation of  $\partial \boldsymbol{\tau} / \partial \mathbf{q}$  and  $\partial \boldsymbol{\tau} / \partial \dot{\mathbf{q}}$ . By differentiating (29) with respect to  $\mathbf{q}$  and  $\dot{\mathbf{q}}$ , it follows

$$\frac{\partial \boldsymbol{\tau}}{\partial \mathbf{q}} = \mathbf{W}^T \frac{\partial \boldsymbol{\tau}_0}{\partial \mathbf{q}} + \frac{\partial \mathbf{W}^T}{\partial \mathbf{q}} \boldsymbol{\tau}_0 = \mathbf{W}^T \frac{\partial \boldsymbol{\tau}_0}{\partial \boldsymbol{\theta}} \mathbf{W} + \mathbf{G}^T \boldsymbol{\tau}_0, \quad (33)$$

$$\frac{\partial \boldsymbol{\tau}}{\partial \dot{\mathbf{q}}} = \mathbf{W}^T \frac{\partial \boldsymbol{\tau}_0}{\partial \dot{\mathbf{q}}} + \frac{\partial \mathbf{W}^T}{\partial \dot{\mathbf{q}}} \boldsymbol{\tau}_0 = \mathbf{W}^T \frac{\partial \boldsymbol{\tau}_0}{\partial \dot{\boldsymbol{\theta}}} \mathbf{W}. \quad (34)$$

Thereby the definition of the Jacobian matrix  $\partial \boldsymbol{\theta} / \partial \mathbf{q} = \partial \dot{\boldsymbol{\theta}} / \partial \dot{\mathbf{q}} = \mathbf{W}$  is used which can be obtained by differentiating (20) with respect to  $\dot{\mathbf{q}}$ . These equations allow to compute the quantities of the original closed loop system  $\partial \boldsymbol{\tau} / \partial \mathbf{q}$ ,  $\partial \boldsymbol{\tau} / \partial \dot{\mathbf{q}}$  from the quantities of the virtual open loop tree-structured system  $\boldsymbol{\tau}_0$ ,  $\partial \boldsymbol{\tau}_0 / \partial \boldsymbol{\theta}$ ,  $\partial \boldsymbol{\tau}_0 / \partial \dot{\boldsymbol{\theta}}$  by using  $\mathbf{W}$ ,  $\mathbf{G}$ .

From (2)–(4), the partial derivative of the state equations  $\mathbf{f}$  with respect to  $\mathbf{u}$  can be expressed as

$$\frac{\partial \mathbf{f}}{\partial \mathbf{u}} = \begin{bmatrix} \frac{\partial \dot{\mathbf{q}}}{\partial \boldsymbol{\tau}} \\ \frac{\partial \ddot{\mathbf{q}}}{\partial \boldsymbol{\tau}} \end{bmatrix} = \begin{bmatrix} \mathbf{O} \\ \frac{\partial \ddot{\mathbf{q}}}{\partial \boldsymbol{\tau}} \end{bmatrix}, \quad (35)$$

$$\frac{\partial \ddot{\mathbf{q}}}{\partial \boldsymbol{\tau}} = \frac{\partial \mathbf{h}}{\partial \boldsymbol{\tau}} = \mathbf{M}^{-1}. \quad (36)$$

In the following, an algorithm to compute  $\mathbf{f}$ ,  $\partial \mathbf{f} / \partial \mathbf{x}$ ,  $\partial \mathbf{f} / \partial \mathbf{u}$  for given  $\mathbf{x}$ ,  $\mathbf{u}$ , i.e.  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ ,  $\boldsymbol{\tau}$  is summarized. It should be noticed that the methods to compute the quantities  $\boldsymbol{\tau}_0$ ,  $\partial \boldsymbol{\tau}_0 / \partial \boldsymbol{\theta}$ ,  $\partial \boldsymbol{\tau}_0 / \partial \dot{\boldsymbol{\theta}}$  for open loop tree-structured systems are already established [11]. We define  $\boldsymbol{\tau}_1(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}})$  as the function that calculates  $\boldsymbol{\tau}_0$  for given  $\boldsymbol{\theta}$ ,  $\dot{\boldsymbol{\theta}}$ ,  $\ddot{\boldsymbol{\theta}}$  by using the inverse dynamics algorithm for open loop tree-structured systems. We also define  $\boldsymbol{\tau}_2(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}})$  as the function that calculates  $\partial \boldsymbol{\tau}_0 / \partial \boldsymbol{\theta}$ ,  $\partial \boldsymbol{\tau}_0 / \partial \dot{\boldsymbol{\theta}}$  for given  $\boldsymbol{\theta}$ ,  $\dot{\boldsymbol{\theta}}$ ,  $\ddot{\boldsymbol{\theta}}$  by using the algorithm proposed in [11] for open loop tree-structured systems.

### Algorithm

1. For a given  $\mathbf{q}$ , find  $\mathbf{p}$  that satisfies the loop closing conditions (9) by using the Newton–Raphson algorithm, then get  $\boldsymbol{\theta} = [\mathbf{q}^T \mathbf{p}^T]^T$ .

2. Compute  $\Phi_q, \Phi_p$  by using Orin's method [17], and compute  $\Phi_{pq}, \Phi_{qq}$  by using Nakamura's method [16]. Calculate  $\mathbf{W}, \mathbf{G}$  by (21) and (23), respectively.
3. Compute  $\dot{\theta}$  corresponding to  $\dot{q}$  by (20).
4. Compute  $\bar{c}(\theta, \dot{\theta}) = \tau_1(\theta, \dot{\theta}, \mathbf{0})$ .
5. Let  $\mathbf{e}_i \in R^{n_0}$  be the  $i$ th unit vector. Compute  $\bar{\mathbf{m}}_i = \tau_1(\theta, \dot{\theta}, \mathbf{e}_i) - \bar{c}$  for  $i = 1$  to  $n_0$ , then calculate the inertia matrix as  $\bar{\mathbf{M}}(\theta) = [\bar{\mathbf{m}}_1 \ \bar{\mathbf{m}}_2 \ \cdots \ \bar{\mathbf{m}}_{n_0}]$ .
6. Compute  $\mathbf{M}(q), c(q, \dot{q})$  by (27) and (28).
7. Compute  $\ddot{q} = \mathbf{M}^{-1}(\tau - c)$ , then get  $\mathbf{f}$  by (2) and (3).
8. Compute  $\ddot{\theta}$  by (22).
9. Compute  $\tau_0 = \tau_1(\theta, \dot{\theta}, \ddot{\theta})$ .
10. Compute  $(\partial \tau_0 / \partial \theta, \partial \tau_0 / \partial \dot{\theta}) = \tau_2(\theta, \dot{\theta}, \ddot{\theta})$ .
11. Compute  $\partial \tau / \partial q, \partial \tau / \partial \dot{q}$  by (33) and (34).
12. Compute  $\partial \ddot{q} / \partial q, \partial \ddot{q} / \partial \dot{q}$  by (31), (32) and compute  $\partial \ddot{q} / \partial \tau$  by (36).
13. Compute  $\partial \mathbf{f} / \partial \mathbf{x}$  by (30) and compute  $\partial \mathbf{f} / \partial \mathbf{u}$  by (35).

The proposed method satisfies the loop closing conditions (9) at all times. However, this method is difficult to apply to redundant actuation systems, i.e.  $n_a > n$ , since in that case such a simple relationship as (29) does not hold anymore.

## 4 Optimal Trajectory Planning Algorithm Based on an Augmented Formulation

In this section, an optimal trajectory planning algorithm that can be applied to the redundant actuation systems is formulated based on the problem formulation 2. The values of  $\bar{\mathbf{f}}, \partial \bar{\mathbf{f}} / \partial \bar{\mathbf{x}}, \partial \bar{\mathbf{f}} / \partial \bar{\mathbf{u}}$  required for optimal control calculation are computed directly from the differential-algebraic equations (8) and (9).

Differentiating (9) twice with respect to time yields

$$\ddot{\Phi}(\theta) = \Phi_{\theta} \ddot{\theta} + (\Phi_{\theta} \dot{\theta})_{\theta} \dot{\theta} = \Phi_{\theta} \ddot{\theta} + \Phi_{\theta\theta} \dot{\theta}^2 = \mathbf{0}. \quad (37)$$

Equations (8) and (37) can be combined in matrix form as

$$\begin{bmatrix} \bar{\mathbf{M}} & \Phi_{\theta}^T \\ \Phi_{\theta} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \lambda \end{bmatrix} + \begin{bmatrix} \bar{c}(\theta, \dot{\theta}) \\ \gamma(\theta, \dot{\theta}) \end{bmatrix} = \begin{bmatrix} \tau_0 \\ \mathbf{0} \end{bmatrix}, \quad (38)$$

where

$$\gamma(\theta, \dot{\theta}) = \Phi_{\theta\theta} \dot{\theta}^2. \quad (39)$$

The vector of accelerations and Lagrange multipliers can be obtained by solving (38) as

$$\begin{bmatrix} \ddot{\theta} \\ \lambda \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{M}} & \Phi_{\theta}^T \\ \Phi_{\theta} & \mathbf{O} \end{bmatrix}^{-1} \begin{bmatrix} \tau_0 - \bar{c} \\ -\gamma \end{bmatrix}. \quad (40)$$

From the upper part of (40), the function  $\bar{\mathbf{h}} : \theta, \dot{\theta}, \tau_0 \mapsto \ddot{\theta}$  in (11) can be defined as

$$\ddot{\theta} = \bar{\mathbf{h}}(\theta, \dot{\theta}, \tau_0) = \mathbf{M}_{11}(\tau_0 - \bar{c}) - \mathbf{M}_{12}\gamma, \quad (41)$$

where we defined  $\mathbf{M}_{ij}$  as

$$\begin{aligned} \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} &= \begin{bmatrix} \bar{\mathbf{M}} & \Phi_\theta^T \\ \Phi_\theta & \mathbf{O} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \bar{\mathbf{M}}^{-1}(\mathbf{E} + \Phi_\theta^T \mathbf{M}_{22} \Phi_\theta \bar{\mathbf{M}}^{-1}) & -\bar{\mathbf{M}}^{-1} \Phi_\theta^T \mathbf{M}_{22} \\ \mathbf{M}_{12}^T & -(\Phi_\theta \bar{\mathbf{M}}^{-1} \Phi_\theta^T)^{-1} \end{bmatrix}. \end{aligned} \quad (42)$$

Next, we consider the partial derivative of  $\bar{\mathbf{f}}$  with respect to  $\bar{\mathbf{x}}$ . From (10) and (11),  $\partial \bar{\mathbf{f}} / \partial \bar{\mathbf{x}}$  can be expressed as

$$\frac{\partial \bar{\mathbf{f}}}{\partial \bar{\mathbf{x}}} = \begin{bmatrix} \frac{\partial \dot{\theta}}{\partial \theta} & \frac{\partial \dot{\theta}}{\partial \dot{\theta}} \\ \frac{\partial \ddot{\theta}}{\partial \theta} & \frac{\partial \ddot{\theta}}{\partial \dot{\theta}} \end{bmatrix} = \begin{bmatrix} \mathbf{O} & \mathbf{E} \\ \frac{\partial \ddot{\theta}}{\partial \theta} & \frac{\partial \ddot{\theta}}{\partial \dot{\theta}} \end{bmatrix}. \quad (43)$$

In order to compute  $\partial \bar{\mathbf{f}} / \partial \bar{\mathbf{x}}$ , the calculation of  $\partial \ddot{\theta} / \partial \theta$  and  $\partial \ddot{\theta} / \partial \dot{\theta}$  is necessary. It is noticed that the following relationship can be proven

$$\frac{\partial \ddot{\theta}}{\partial \theta} = -\mathbf{M}_{11} \frac{\partial \tau_0}{\partial \theta}, \quad (44)$$

$$\frac{\partial \ddot{\theta}}{\partial \dot{\theta}} = -\mathbf{M}_{11} \frac{\partial \tau_0}{\partial \dot{\theta}}, \quad (45)$$

where  $\mathbf{M}_{11}$  is defined in (42), i.e.  $\mathbf{M}_{11} = \bar{\mathbf{M}}^{-1} \{ \mathbf{E} - \Phi_\theta^T (\Phi_\theta \bar{\mathbf{M}}^{-1} \Phi_\theta^T)^{-1} \Phi_\theta \bar{\mathbf{M}}^{-1} \}$ .

#### Proof of Equations (44) and (45)

Equation (38) can be rewritten as

$$\hat{\mathbf{M}}(\theta) \ddot{\theta} + \hat{\mathbf{c}}(\theta, \dot{\theta}) = \hat{\tau}, \quad (46)$$

where

$$\ddot{\theta} = \begin{bmatrix} \ddot{\theta} \\ \lambda \end{bmatrix}, \quad \hat{\mathbf{M}} = \begin{bmatrix} \bar{\mathbf{M}} & \Phi_\theta^T \\ \Phi_\theta & \mathbf{O} \end{bmatrix}, \quad \hat{\mathbf{c}} = \begin{bmatrix} \bar{\mathbf{c}} \\ \gamma \end{bmatrix}, \quad \hat{\tau} = \begin{bmatrix} \tau_0 \\ \mathbf{0} \end{bmatrix}. \quad (47)$$

Then, (40) can be expressed as

$$\ddot{\theta} = \hat{\mathbf{M}}(\theta)^{-1} \{ \hat{\tau} - \hat{\mathbf{c}}(\theta, \dot{\theta}) \}. \quad (48)$$

Since (46) and (48) have the same form as the equations of motion of open loop systems, we can derive the following equations by using the same manner as presented in [11]

$$\frac{\partial \ddot{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} = -\hat{\mathbf{M}}^{-1} \frac{\partial \hat{\boldsymbol{\tau}}}{\partial \boldsymbol{\theta}}, \quad (49)$$

$$\frac{\partial \ddot{\boldsymbol{\theta}}}{\partial \dot{\boldsymbol{\theta}}} = -\hat{\mathbf{M}}^{-1} \frac{\partial \hat{\boldsymbol{\tau}}}{\partial \dot{\boldsymbol{\theta}}}. \quad (50)$$

From (42) and (47), (49) can be written as

$$\begin{bmatrix} \frac{\partial \ddot{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} \\ \frac{\partial \ddot{\boldsymbol{\theta}}}{\partial \lambda} \\ \frac{\partial \ddot{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} \end{bmatrix} = - \begin{bmatrix} \bar{\mathbf{M}} & \Phi_{\theta}^T \\ \Phi_{\theta} & \mathbf{O} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \boldsymbol{\tau}_0}{\partial \boldsymbol{\theta}} \\ \frac{\partial \boldsymbol{\tau}_0}{\partial \lambda} \\ \mathbf{O} \end{bmatrix} = - \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} \frac{\partial \boldsymbol{\tau}_0}{\partial \boldsymbol{\theta}} \\ \frac{\partial \boldsymbol{\tau}_0}{\partial \lambda} \\ \mathbf{O} \end{bmatrix}. \quad (51)$$

The upper part of (51) is equal to (44). In the same way, (45) can be derived from (50).

In order to solve the optimal trajectory planning problem the partial derivative of the state equations  $\bar{\mathbf{f}}$  with respect to  $\bar{\mathbf{u}}$  is necessary. This can be expressed from (10), (11), and (41) as

$$\frac{\partial \bar{\mathbf{f}}}{\partial \bar{\mathbf{u}}} = \begin{bmatrix} \frac{\partial \dot{\boldsymbol{\theta}}}{\partial \boldsymbol{\tau}_0} \\ \frac{\partial \ddot{\boldsymbol{\theta}}}{\partial \boldsymbol{\tau}_0} \end{bmatrix} = \begin{bmatrix} \mathbf{O} \\ \frac{\partial \ddot{\boldsymbol{\theta}}}{\partial \boldsymbol{\tau}_0} \end{bmatrix}, \quad (52)$$

$$\frac{\partial \ddot{\boldsymbol{\theta}}}{\partial \boldsymbol{\tau}_0} = \frac{\partial \bar{\mathbf{h}}}{\partial \boldsymbol{\tau}_0} = \mathbf{M}_{11}. \quad (53)$$

In the following, an algorithm to compute  $\bar{\mathbf{f}}$ ,  $\partial \bar{\mathbf{f}} / \partial \bar{\mathbf{x}}$ ,  $\partial \bar{\mathbf{f}} / \partial \bar{\mathbf{u}}$  for given  $\bar{\mathbf{x}}, \bar{\mathbf{u}}$ , i.e.  $\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \boldsymbol{\tau}_0$  is summarized. As in the previous section, we use the function  $\boldsymbol{\tau}_1(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}})$  that calculates  $\boldsymbol{\tau}_0$  for given  $\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}$  by using the inverse dynamics algorithm for open loop tree-structured systems, and the function  $\boldsymbol{\tau}_2(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}})$  that calculates  $\partial \boldsymbol{\tau}_0 / \partial \boldsymbol{\theta}$ ,  $\partial \boldsymbol{\tau}_0 / \partial \dot{\boldsymbol{\theta}}$  for given  $\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}$  by using the algorithm proposed in [11] for open loop tree-structured systems.

### Algorithm

1. Compute  $\Phi_{\theta}$  by using Orin's method [17], and compute  $\Phi_{\theta\theta}$  by using Nakamura's method [16].
2. Compute  $\boldsymbol{\gamma}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$  by (39).
3. Compute  $\bar{\mathbf{c}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \boldsymbol{\tau}_1(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \mathbf{0})$ .
4. Define  $\mathbf{e}_i \in R^{n_0}$  as the  $i$ th unit vector. Compute  $\bar{\mathbf{m}}_i = \boldsymbol{\tau}_1(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \mathbf{e}_i) - \bar{\mathbf{c}}$  for  $i = 1$  to  $n_0$ , then calculate the inertia matrix as  $\bar{\mathbf{M}}(\boldsymbol{\theta}) = [\bar{\mathbf{m}}_1 \ \bar{\mathbf{m}}_2 \ \cdots \ \bar{\mathbf{m}}_{n_0}]$ .
5. Compute  $\mathbf{M}_{11}$  by (42).
6. Compute  $\ddot{\boldsymbol{\theta}}$  by solving (38), then get  $\bar{\mathbf{f}}$  by (10) and (11).
7. Compute  $(\partial \boldsymbol{\tau}_0 / \partial \boldsymbol{\theta}, \partial \boldsymbol{\tau}_0 / \partial \dot{\boldsymbol{\theta}}) = \boldsymbol{\tau}_2(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}})$ .
8. Compute  $\partial \ddot{\boldsymbol{\theta}} / \partial \boldsymbol{\theta}$ ,  $\partial \ddot{\boldsymbol{\theta}} / \partial \dot{\boldsymbol{\theta}}$  by (44), (45) and compute  $\partial \ddot{\boldsymbol{\theta}} / \partial \boldsymbol{\tau}_0$  by (53).
9. Compute  $\partial \bar{\mathbf{f}} / \partial \bar{\mathbf{x}}$  by (43) and compute  $\partial \bar{\mathbf{f}} / \partial \bar{\mathbf{u}}$  by (52).

The method presented here can also be applied to redundant actuation systems without any problems. However, it suffers potentially from so called constraint violation since it uses the constraint formulation on acceleration level given by (37) instead of the constraint equation on position level given by (9). Though it is not proven mathematically, this method may give accurate results in most practical applications by combining it with a constraint stabilization technique, e.g. Baumgarte's method [2].

## 5 Numerical Examples

### 5.1 Non-redundant Actuation System

In this section, the optimal trajectory planning algorithm using the coordinate partitioning and embedding techniques from Sect. 3 is demonstrated for computing the minimum energy trajectory of a closed loop robot manipulator as shown in Fig. 4. This system is a non-redundant actuation system, featuring 3 degree of freedom and 3 actuators. We transform the closed loop system to a virtual open loop tree-structured system by cutting at the joint marked with  $\times$  in Fig. 4. The kinematical parameters (Denavit–Hartenberg parameters [6]) and the dynamical parameters are summarized in Tables 1 and 2, respectively. The vector of generalized coordinates  $\mathbf{q}$  and the vector of joint variables of the virtual open loop tree-structured system  $\boldsymbol{\theta}$  are

$$\mathbf{q} = [q_1 \ q_2 \ q_3]^T = [\theta_1 \ \theta_2 \ \theta_3]^T, \quad (54)$$

$$\boldsymbol{\theta} = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5]^T. \quad (55)$$

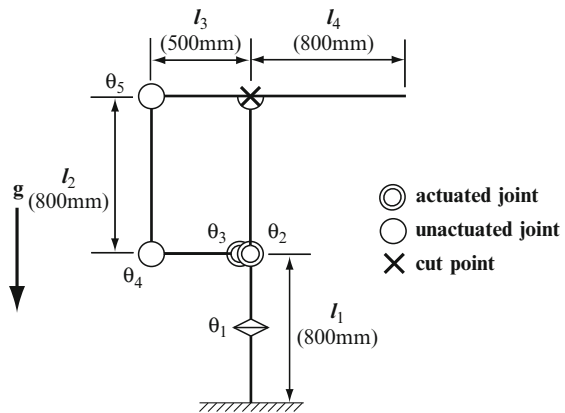


Fig. 4 A closed loop robot manipulator, see also Fig. 7

**Table 1** Kinematical parameters of the closed loop robot manipulator

Link	Length	Twist	Offset	Angle
$i$	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
1	0	$0^\circ$	0	$\theta_1$
2	0	$90^\circ$	0	$\theta_2$
3	0	$90^\circ$	0	$\theta_3$
4	$l_3$	$0^\circ$	0	$\theta_4$
5	$l_2$	$0^\circ$	0	$\theta_7$

**Table 2** Dynamical parameters of the closed loop robot manipulator

Link	Mass (kg)	Moment of inertia (kg·m <sup>2</sup> )	Center of gravity (m)
$i$	$m_i$	${}^i\hat{\mathbf{I}}_i = \text{diag}[I_x, I_y, I_z]$	${}^i\hat{\mathbf{s}}_i = [x, y, z]^T$
1	10.0	(1.4100, 1.4100, 1.4100)	(0.00, 0.00, -0.25)
2	5.4	(0.0023, 0.2891, 0.2891)	(0.40, 0.00, 0.00)
3	3.375	(0.0014, 0.0710, 0.0710)	(0.25, 0.00, 0.00)
4	5.4	(0.0023, 0.2891, 0.2891)	(0.40, 0.00, 0.00)
5	9.0	(0.0037, 1.2376, 1.2376)	(0.65, 0.00, 0.00)

For this problem, we can find the following simple relationships by geometric inspection

$$\theta_4 = q_2 - q_3, \quad (56)$$

$$\theta_5 = -\pi - q_2 + q_3. \quad (57)$$

Hence, the Jacobian matrix and the Hessian matrix become

$$\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}, \quad \mathbf{G} = \mathbf{O}. \quad (58)$$

The cost function is defined here as control energy

$$J = \int_0^{t_f} \mathbf{u}^T \mathbf{u} \, dt, \quad (59)$$

where  $\mathbf{u} = [\tau_1 \ \tau_2 \ \tau_3]^T$ . As an example, we consider the problem to find a trajectory from the initial configuration  $\mathbf{q}(0) = [0^\circ \ 30^\circ \ 180^\circ]^T$  to the desired configuration  $\mathbf{q}(t_f) = [60^\circ \ 90^\circ \ 240^\circ]^T$  and vanishing velocity at both ends i.e.  $\dot{\mathbf{q}}(0) = \dot{\mathbf{q}}(t_f) = \mathbf{0}$ . Since the optimal solution depends on the final time  $t_f$ , we solve the problem for different final times  $t_f$ . Figure 5 shows the minimal value of  $J$ , i.e. the energy, for different final times  $t_f$ .



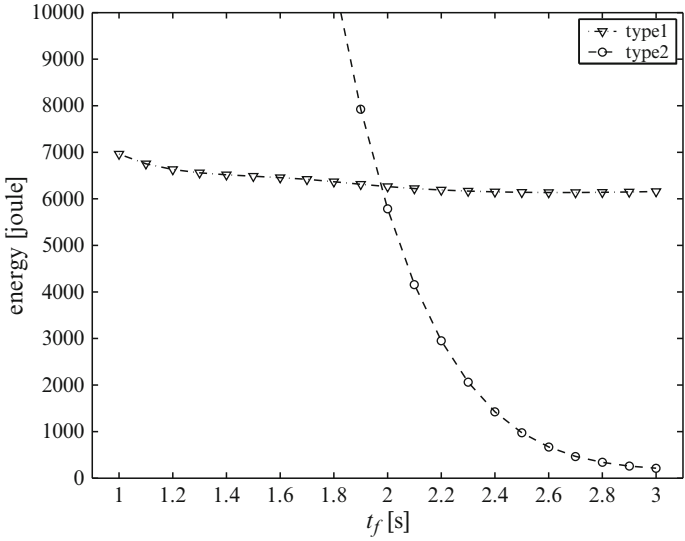


Fig. 5 Minimum energy for different final times  $t_f$

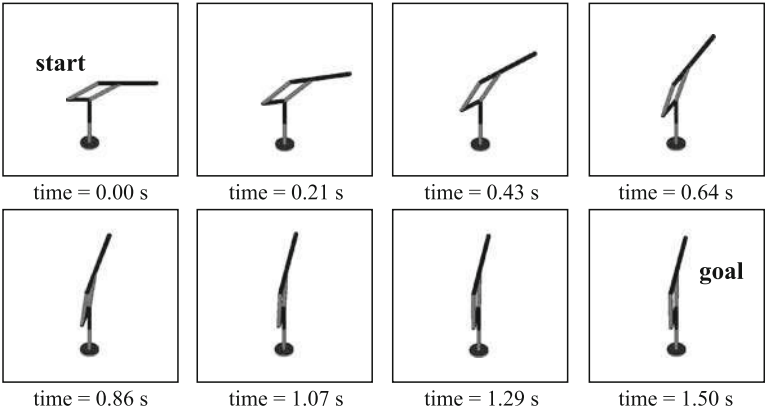


Fig. 6 Optimal trajectory of the closed loop robot manipulator (type 1,  $t_f = 1.5$  s)

For this problem, two types of trajectories that satisfy the necessary condition for optimality are found. The type 1 trajectory is the one that connects the initial and final configuration linearly as shown in Fig. 6. The type 2 trajectory is the one in which the links fall down in the gravity direction at first while their arms are folding up, then fling up to the desired configuration by using the inertial influence as shown in Fig. 7. From Fig. 5, it is seen that the minimum energy trajectory switches around  $t_f = 2$  s, that is, the type 1 trajectory requires least energy if  $t_f < 2$  s and the type 2 trajectory requires least energy if  $t_f > 2$  s. In a conventional application, a kind of linear motion like the type 1 trajectory is usually used. However, this simulation

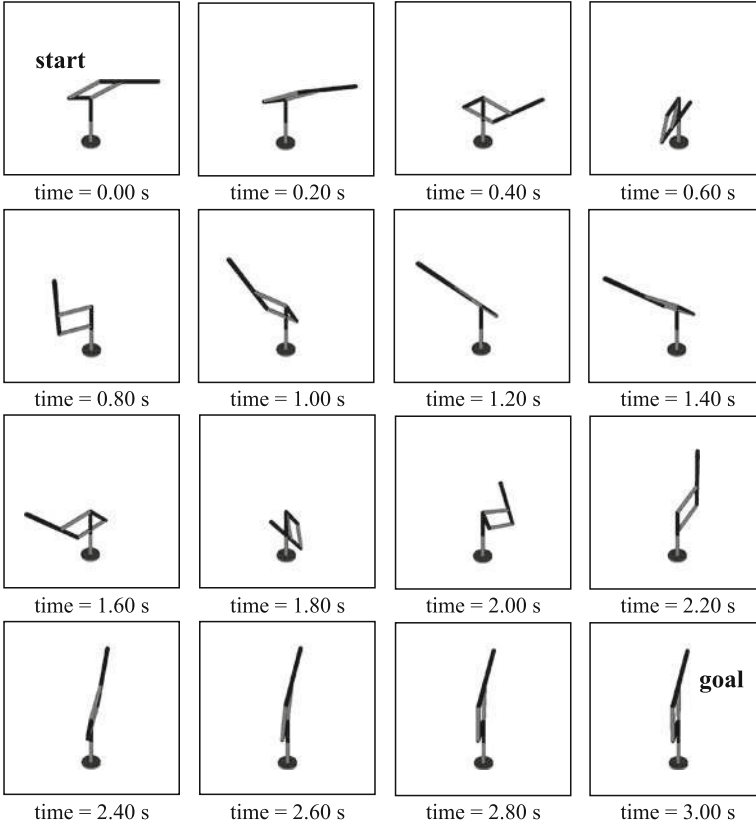


Fig. 7 Optimal trajectory of the closed loop robot manipulator (type 2,  $t_f = 3.0$  s)

shows the possibility that we can reduce the energy consumption dramatically by selecting a more complex motion like the type 2 trajectory. Figure 8 shows the trajectory of joint variables for the case of type 2 and  $t_f = 3$  s.

## 5.2 Redundant Actuation System

The optimal trajectory planning algorithm based on the augmented formulation presented in Sect. 4 is demonstrated for computing the minimum energy trajectory of a planar parallel link manipulator as shown in Fig. 9. This system is a redundant actuation system, because it has 3 degrees of freedom and 4 actuators. We transform the closed loop system to a virtual open loop tree-structured system by cutting at the joint marked with  $\times$  in Fig. 9. The kinematical and dynamical parameters are given in Tables 3 and 4, respectively.

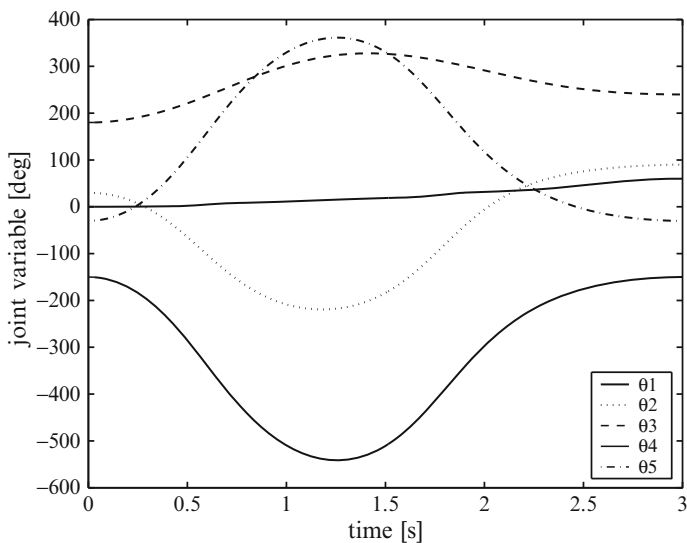


Fig. 8 Joint variables versus time (type 2,  $t_f = 3.0$  s)

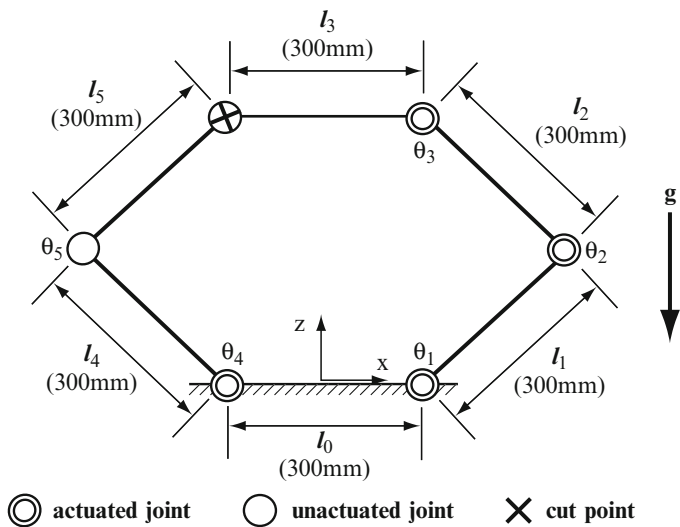


Fig. 9 A planar parallel link manipulator

In this example the cost function is again defined as control energy

$$J = \int_0^{t_f} \mathbf{u}^T \mathbf{u} dt, \quad (60)$$

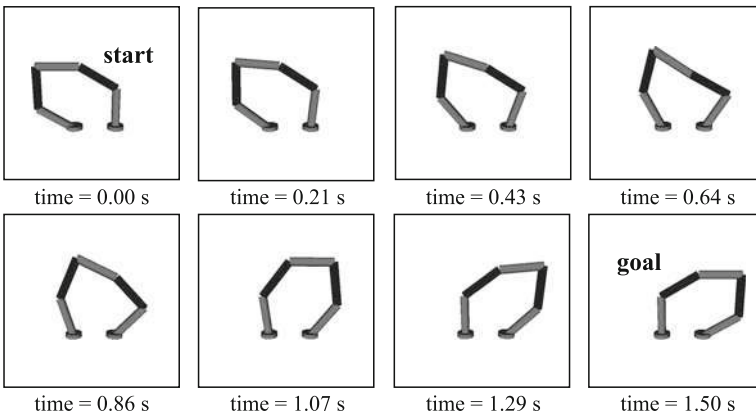
where  $\mathbf{u} = [\tau_1 \ \tau_2 \ \tau_3 \ \tau_4]^T$ . As an example, we consider the problem to find a trajectory from the initial configuration  $\boldsymbol{\theta}(0) = [90^\circ \ 60^\circ \ 30^\circ \ -30^\circ \ -60^\circ]^T$  to the

**Table 3** Kinematical parameters of the planar parallel link manipulator

Link $i$	Length $a_i$	Twist $\alpha_i$	Offset $d_i$	Angle $\theta_i$
1	$l_0/2$	$90^\circ$	0	$\theta_1$
2	$l_1$	$0^\circ$	0	$\theta_2$
3	$l_2$	$0^\circ$	0	$\theta_3$
4	$-l_0/2$	$90^\circ$	0	$180^\circ + \theta_4$
5	$l_4$	$0^\circ$	0	$\theta_5$

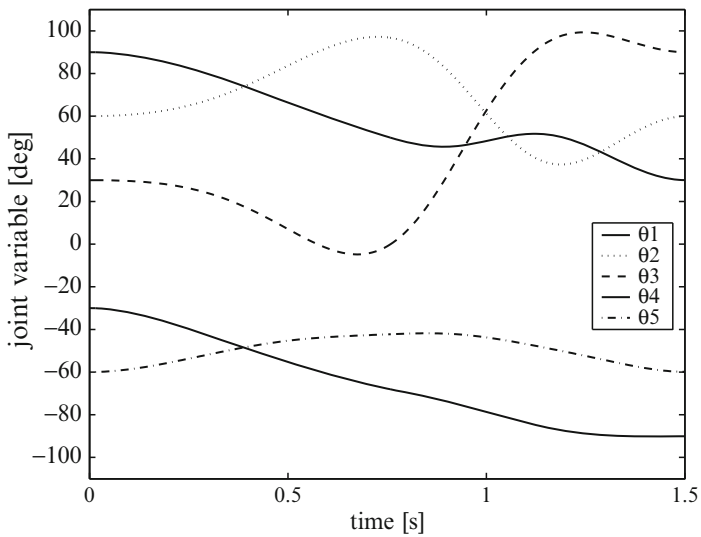
**Table 4** Dynamical parameters of the planar parallel link manipulator

Link $i$	Mass (kg) $m_i$	Moment of inertia (kg-m <sup>2</sup> ) ${}^i\hat{\mathbf{I}}_i = \text{diag}[I_x, I_y, I_z]$	Center of gravity (m) ${}^i\hat{\mathbf{s}}_i = [x, y, z]^T$
1	3.0	(0.05, 0.80, 0.80)	(0.15, 0.00, 0.00)
2	3.0	(0.05, 0.80, 0.80)	(0.15, 0.00, 0.00)
3	3.0	(0.05, 0.80, 0.80)	(0.15, 0.00, 0.00)
4	3.0	(0.05, 0.80, 0.80)	(0.15, 0.00, 0.00)
5	3.0	(0.05, 0.80, 0.80)	(0.15, 0.00, 0.00)

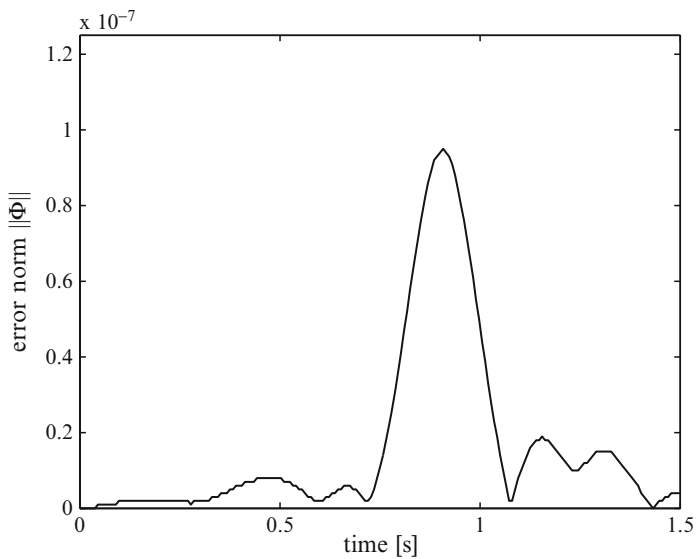


**Fig. 10** Optimal trajectory of the planar parallel link manipulator ( $t_f = 1.5$  s)

final configuration  $\boldsymbol{\theta}(t_f) = [30^\circ \ 60^\circ \ 90^\circ \ -90^\circ \ -60^\circ]^T$  and  $\dot{\boldsymbol{\theta}}(0) = \dot{\boldsymbol{\theta}}(t_f) = \mathbf{0}$ . The final time is fixed and defined as  $t_f = 1.5$  s. Baumgarte's method [2] is used to stabilize the loop closing conditions. The obtained optimal trajectory is shown in Fig. 10. Figure 11 shows the trajectory of joint variables. Figure 12 shows the error norm of the loop closing condition, i.e.  $\|\boldsymbol{\Phi}\|$ . As one can see, the error is at most  $10^{-7}$ , therefore, the method is considered here as accurate enough for practical use.



**Fig. 11** Joint variables versus time ( $t_f = 1.5$  s)



**Fig. 12** Constraint violation versus time ( $t_f = 1.5$  s)

## 6 Conclusions

In this paper, a general purpose optimal trajectory planning algorithm for multibody systems is proposed. Two methods for closed loop systems were developed by extending the algorithm for open loop systems. The algorithm using coordinate

partitioning and embedding techniques is exact in the sense that it satisfies the loop closing conditions at all times but it is difficult to apply to redundant actuation systems. On the other hand, the algorithm based on the augmented formulation can be applied to the redundant actuation systems quite easily but it can not be guaranteed that the loop closing conditions are always satisfied. Hence, one should check the results obtained by the second algorithm properly. These algorithms are considered to be useful since they allow to conduct optimal trajectory calculation for arbitrary closed loop multibody systems from the kinematical and dynamical parameters, and the cost function. The efficiency of the methods is demonstrated using two closed loop robot manipulators.

**Acknowledgements** The authors would like to thank Prof. N. Shimizu (Iwaki Meisei University) for warm encouragement and help.

## References

1. Agrawal OP, Xu Y (1994) On the global optimum path planning for redundant space manipulator. *IEEE T Syst Man Cyb* 24(9):1306–1316
2. Baumgarte J (1972) Stabilization of constraints and integrals of motion in dynamical systems. *Comput Meth Appl Mech Eng* 1:1–16
3. Bessonnet G, Chesse S, Sardain P (2004) Optimal gait synthesis of a seven-link planar biped. *Int J Robot Res* 23(10–11):1059–1073
4. Bobrow JE (1988) Optimal robot path planning using the minimum-time criterion. *IEEE J Robot Autom* 4(4):443–450
5. Bryson AE Jr, Ho YC (1975) *Applied optimal control*. Hemisphere Publishing Corporation, Washington
6. Craig JJ (2004) *Introduction to robotics, mechanics and control*, Prentice Hall, Englewood Cliffs, NJ
7. Dubowsky S, Shiller Z (1985) *Optimal dynamic trajectories of robotic manipulators. Theory and practice of robots and manipulators*. MIT Press, Cambridge: 133–143
8. Eberhard P, Schiehlen W (2006) Computational dynamics of multibody systems: history, formalisms, and applications. *J Comput Nonlin Dyn* 1:3–12
9. Fernandes C, Gurfits L, Li ZX (1994) Near-optimal nonholonomic motion planning for a system of coupled rigid bodies. *IEEE T Automat Contr* 30(3):450–463
10. Geering HP, Guzzella L, Hepner SAR, Onder CH (1986) Time-optimal motions of robots in assembly tasks. *IEEE T Automat Contr* 31(6):512–518
11. Iwamura M, Eberhard P, Schiehlen W, Seifried R (2009) A general purpose optimal trajectory planning algorithm for multibody systems. In: *Proceedings of the IASTED international conference on modeling, identification, and control*, Innsbruck, Austria, 16–18 February
12. Iwamura M, Ozaki H, Mohri A (2003) A new optimal control algorithm and its application to shortest path planning for cars. In: *Proceedings of the IASTED international conference on intelligent systems and control*, Salzburg, Austria, 25–27 June
13. Kim B, Shin K (1985) Suboptimal control of industrial manipulators with a weighted minimum time-fuel criterion. *IEEE T Automat Contr* 30(1):1–10
14. Luh JYS, Zheng Y (1985) Computation of input generalized force for robots with closed kinematic chain mechanisms. *IEEE J Robot Autom* 1:95–103
15. Nakamura Y (1991) *Advanced robotics, redundancy and optimization*. Addison-Wesley, Reading, MA

16. Nakamura Y, Yokokohji Y, Hanafusa H, Yoshikawa T (1986) Unified recursive formulation of kinematics and dynamics of robot manipulators. In: Proceedings of Japan–U.S.A. symposium on flexible automation, Osaka, Japan, July
17. Orin DE, Schrader WW (1984) Efficient computation of the Jacobian for robot manipulators. *Int J Robot Res* 3(4):66–75
18. Pontryagin L, Boltiansky V, Gamkrelitze A, Mishchenko E (1962) The mathematical theory of optimal processes. Wiley, New York
19. Sakawa Y, Shindo Y (1980) On global convergence of an algorithm for optimal control. *IEEE T Automat Contr* 25(6):1149–1153
20. Shin K, McKay ND (1986) A dynamic programming approach to trajectory planning of robotic manipulators. *IEEE T Automat Contr* 31(6):491–500
21. Smith DA (1973) Reaction force analysis in generalized machine systems. *J Eng Ind Trans ASME (Ser B)* 95:617–623

# Real-Time Simulation of Extended Vehicle Drivetrain Dynamics

Ralf U. Pfau and Thomas Schaden

**Abstract** For the virtual engine development, testing and calibration, it is advantageous to use the same physical model on different platforms. Due to the complexity of the model and its evaluation one has to cope with severe evaluation restrictions on the realtime platform. For coupled problems which include an electrical system, the equilibrium conditions include algebraic constraints. Hence it is not sufficient to use only an explicit time integration scheme. We extend an explicit scheme to a mixed scheme such that the overall performance per time step still is below the timing constraint of the realtime platform for reasonable complex model with electrical system.

## 1 Introduction

Driven by increasing computing power and by increasing pressure for shorter product cycles more and more of the construction process for new cars is accompanied by and interwoven with computer simulation. There are different tools and tool chains for the construction, simulation and evaluation of designs and systems. Though the trend is to couple the tools and simulations for a fully virtual engine, powertrain and vehicle simulation which allows to take also multi-physical aspects and interactions into consideration. Though, increasing computing power and available memory supports this, analysis and refinements for the coupled problems are needed to take full advantage of the coupled problem setting and to accurately and efficiently solve the coupled problem.

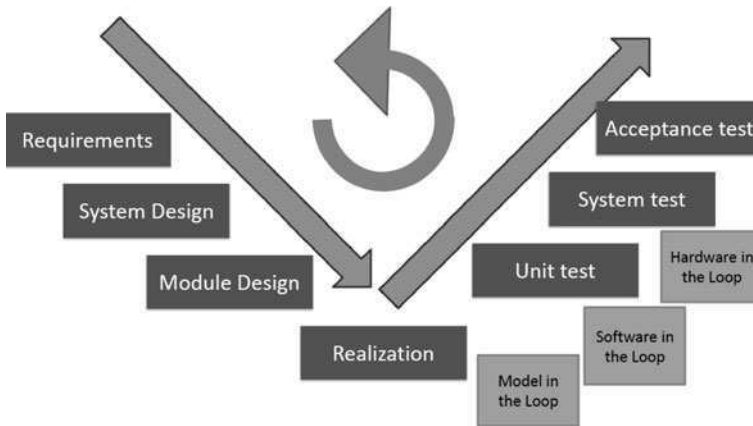
Additionally, as all modelling and simulation inhabits some compromise with respect to details of the physics and accuracy, one also wants to have a smooth

---

R.U. Pfau (✉)  
MathConsult, Altenberger Str. 69, 4040 Linz, Austria  
e-mail: [ralf.pfau@mathconsult.co.at](mailto:ralf.pfau@mathconsult.co.at)

T. Schaden  
AVL List GmbH, Hans-List-Platz 1, 8040 Graz, Austria  
e-mail: [thomas.schaden@avl.com](mailto:thomas.schaden@avl.com)





**Fig. 1** V development scheme (adapted to our situation)

transition from pure software models to hardware. This allows to start with simple models, e.g. with data from more detailed computer simulations, going to refinements within the models, updating parameters from prototype measurements, but also coupling the software model with testbed system and replacing step by step the computer model by hardware, e.g. engine, ECU, transmissions. See Fig. 1 for a sketch of the iterative approach used in the well known V development model (see [22, 28, 36] for integrated development processes).

The first simulation in this style started at the end of the eighties and beginning of the nineties ([20] and the references therein). Where one used then the testbed and measurement for a verification of the computer simulation before going to the assembled system.

It is easier to accomplish realtime simulation with smaller sized systems where a single component or vehicle part is modelled. The tendency is then to be more detailed in the description of the model to reduce the modelling error as much as possible, but also to detect failures within a component.

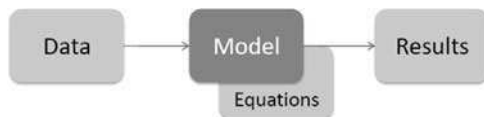
On the other hand, to put a complex model describing a full vehicle behaviour on realtime system, one often employed reduced model, reduced order models and approximation which are parametrized from more detailed off-line simulations or measurements. It is easier to achieve realtime with reduced models, but as the parametrization usually results from standard working conditions, the accuracy may deteriorate if the calculation runs into critical conditions. To verify, if some failure or critical situation results from the model or from the actual configuration or data under investigation, a smooth transition and verification with similar calculation results in the off-line situation is advantageous. Additionally, in the off-line situation, the model may be refined and investigated, if a refinement of the model (the simulation) or the data (change in the hardware, configuration) is needed.

To use the model in different situations, one can either use equation driven or data driven approaches. In case of equation driven approach, one utilizes a model



**Fig. 2** Equation driven stages for simulation

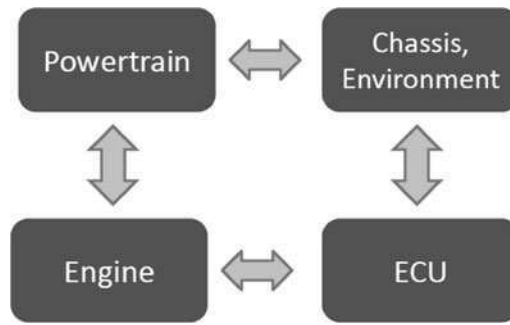
**Fig. 3** Data driven stages  
(fixed equations in  
components)



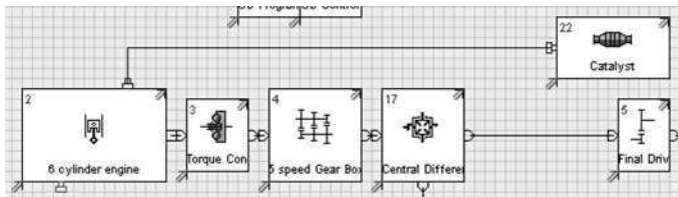
based design, automatic code generation, equation reduction with computer algebra systems, e.g. based on a Modelica description of the components, see Fig. 2 for the stages [8, 10, 24, 33, 38]. Hence the emphasis is on description of the system by equations. On the other hand, with a data driven approach, one has a verified and tested model with a equation system based on physical modelling. The difference between specific configurations is then in the parametrization of the model and components which has to cover sufficiently large domain, see Fig. 3. The challenge then is in the development of those data driven simulation environment whereas the usage and utilization in an industrial context is easier compared to an equation driven approach. In contrast to the equation driven approach where more knowledge (and computer algebra) is used for generating an efficient executable, more has to be done for generating an executable for the data driven simulation. On the other hand, one can consider specific model characteristics and the equation system for an efficient, adapted solution algorithm. As noted in [15] the linear algebra implementation is a critical point for the efficiency, even more for realtime simulations. Specialized numerical methods have been developed for constrained multibody system and the solution of the DAE system with adapted methods utilizing specific approximations of the Jacobian, leading to mixed integration methods [1, 6, 7, 11, 29, 31, 32].

On the testbed system 1 ms simulation time has to be calculated within less then 1 ms CPU time. Normally one has some soft constraint of 0.5 ms CPU time for the simulation of 1 ms simulation time as also the operation system and IO needs CPU time, but a time failure now and then can be coped with as long as some sufficiently accurate data is available within 1 ms. On the testbed system either all can be in software (SIL = software-in-the-Loop) or some parts in hardware (HIL = hardware in the loop), see [8, 24, 28, 34] for experiences and results of other HIL simulations and couplings, starting with smaller system and reduced complexity within the vehicle. But each part on the testbed system communicates with the others via standard channels which are similar in a vehicle and each part does not know whether the other parts it communicates with is software or hardware (see Fig. 4). With respect to system coordination, one has to expect and coop with similar problems and effects as for co-simulation coupling of systems.

In the beginning, it was not possible to run the full multibody system on the realtime environment. Therefore model reduction, reduced models and similar



**Fig. 4** Different parts communicate via standard channels. Parts may be in software or hardware



**Fig. 5** Powertrain example, clip from an AVL CRUISE example model with mechanical connections

approaches have been employed [5, 9, 24, 30]. Also, the complexity of the system has been reduced by restriction on relevant subsystems of the vehicle [21, 33, 35] and the interaction with the other parts have been approximated.

Still to have the full vehicle modelled as elastic multibody system with additional parts and actors is beyond the capacity of the modern HIL systems. In this paper our emphasis is on the powertrain dynamics. That is the interaction and interplay of different components in the car from engine to the tires to yield to overall behaviour and different characteristics of the (assembled) car. An accurate and efficient simulation of the powertrain dynamics is an important tool in the engineering application and tuning of control units. To achieve this, the component model has to be sufficiently detailed and accurate to include all needed physical effects.

The program CRUISE is developed by AVL List for the simulation of the powertrain dynamics of vehicles with respect to driving performance and fuel consumption (Fig. 5). The main array of application has been used the off-line simulation of the vehicle to calculate characteristics. It is possible to include hardware by measurement and setting the measured curves in the components of the model. Some progress has been made recently for the real-time simulation of the mechanical system [27, 36]. The next step is to include advanced electrical components into the driveline. Our focus is on the analyses of and changes for real-time simulation of the extended powertrain (or driveline) systems.

The standard driveline system is a multibody system with some specific characteristics as one has mainly rotational DOFs in the system. Besides the

mechanical behaviour, the properties of the combined systems is crucially influenced by the different controller units which act on the driveline and influence the performance, e.g. through gear change, opening/closing of clutches and similar. For the development, testing and evaluation of novel powertrain concepts additionally electrical components have to be integrated into the simulation to be able to simulate and evaluate hybrid vehicle concepts. This is also important from the aspect of controller adjustment and parametrization as more and more control units are needed for new and hybrid cars whose calibration can efficiently be done only in the simulation.

An example for a successful application of the combined simulation is the AVL Turbohybrid [13]. Downsizing and downspeeding is combined with turbo charging, direct injection and mild hybridization such that driveability as well as fuel consumption for given test cycles have been reduced. It has been possible to test and adjust the different components to achieve a successful test design which fulfills the partly opposite functional requirements [36].

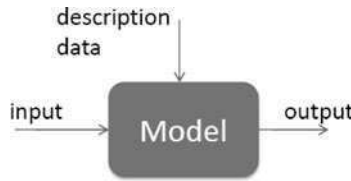
In this paper we concentrate on standard office simulation (abbreviated by *office*) which can be utilized for in-depth investigations and on realtime integration on testbed systems (*realtime*). As noted in [15] not only the model, but also actual implementation of the numeric and the efficient implementation of the numerical linear algebra is critical for the performance. Additionally, for realtime applications with an emphasis on larger multibody systems specific time integrators have been investigated which reduces to linear implicit problems (see [1, 6]). In our case, the interplay between the mechanic and the non-mechanic physics is stronger.

## 1.1 Notations

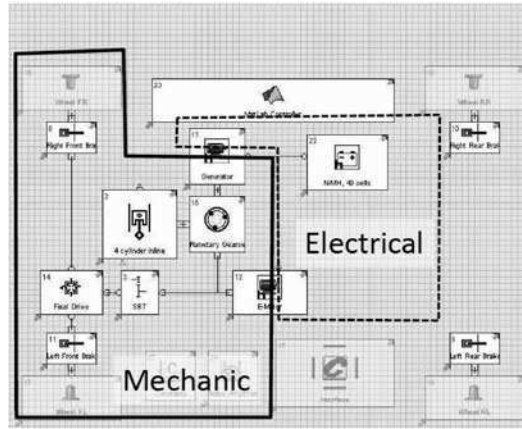
$x, v, a$	mechanical state with way $x$ , velocity $v = \dot{x}$ , acceleration $a = \ddot{x}$
$m$	moments in the system.
$z$	(continuous) control variable, e.g. load signal to the engine
$d$	discrete value in the system, e.g. gear, slip-stick state
$v$	voltage in the electrical system
$i$	current in the component
$t$	time of the system. derivative with respect to time is denoted by $\cdot$
$M$	(diagonal) mass matrix
$F$	(external, internal) forces, torques in the system

## 2 Powertrain System and Vehicle Dynamics

Although the emphasis is on the powertrain system, components which have an major effect on the driveline and driving performance are modelled internally more in details to cover more effects. Other components may include more simplifications.



**Fig. 6** Component structure. The description data may determine the internal complexity of the component and includes the parameterization of the component, e.g. physical constants, measurements



**Fig. 7** Model for realtime system with marked domains for mechanic and electrical system

The actual behaviour of the component is then described by physical constants, measurements and/or simulations.

The vehicle is modelled as blocks, see Fig. 6 which is similar as in Matlab/Simulink or Modelica based systems. Each component can be described as an input-output system with different connections and description data, see Fig. 6 and [2]. More details can be found in [2, 19] or more generally in [37, 39]. The combination of the blocks and the connections yields then the vehicle (see Fig. 7) and an equation system for the vehicle which may be an ODE or DAE.

The internal structure of the components may vary from table interpolation to complex physical models or approximation and simplified models. The parametrization may come from the model, physical constants, measurements, other calculations or parameter identifications. The components may include state events and switches, e.g. of the gear, slip-stick phenomena and similar.

## 2.1 Multibody System

The mechanical system is mainly described through rotational degrees of freedom and leads to an equation system of the form  $F = Ma$  with the diagonal mass

matrix  $M$ . The left hand side forces  $F$  are calculated by the components and may depend on all of the state ( $x, v$ , controls, voltage, current, ...). Additionally, the moments between the components within the system are added as unknowns to be able to specifying moment which depend on the state [19].

This leads to the equation system for the mechanical part

$$\begin{pmatrix} M & B \\ C & D \end{pmatrix} \cdot \begin{pmatrix} a \\ m \end{pmatrix} = \begin{pmatrix} 0 \\ F \end{pmatrix} \quad (1)$$

where  $F$  includes now the moments calculated in the components which acts on the mechanic as well as the losses.  $M$  is the diagonal mass matrix,  $B$  describes the acting of the moments between the components on the components,  $C$  describes the constraints between the mechanical parts and  $D$  the relation between the moments. The coupling between the components can either by via constraints (in the  $C$  part) or via moments. Due to the character of the connections, the last rows split into  $Ca = 0$  and  $Dm = F$ .

*Remark.*

- The equation  $Ca = 0$  is also utilized to reduce the number of degrees of freedom in the system from the total number of DOFs to a reduced number of independent DOFs for the time integration.
- Changes in the system, e.g. gear change, opening and closing of connections, change the matrix parts  $C$  and  $D$ , but not  $M$  and  $F$ .

For the standard time integration the calculation of the inverse of

$$S := \begin{pmatrix} M & B \\ C & D \end{pmatrix} \quad (2)$$

has to be calculated (and stored) after each state event/change in the matrices. This is not critical for the normal time integration.

But under realtime conditions, the inversion of  $S$  is time critical. Therefore a special matrix decomposition based on Schur complement is used. Instead of calculation and usage of  $S^{-1}$  the following is used:

$$(D - CM^{-1}B)m = F \quad (3)$$

$$a = -M^{-1}Bm. \quad (4)$$

As the matrix  $M$  is diagonal, the inversion is cheap. The decomposition and variable management can be calculated beforehand. Also, a sparsity pattern for  $B$  is calculated and used for fast matrix-vector multiplication. Changes in the system now influence  $\tilde{S} := (D - CM^{-1}B)$ . The inverse of  $\tilde{S}$  now has to be calculated/updated after each state event. But this results in an order of magnitude smaller CPU time compared to the inversion of  $S$ .

## 2.2 Control Units

Besides mechanical properties like inertia, ratios and losses, the overall behaviour of the vehicle is determined by the control units acting on the driveline and changing it. The tuning and calibration of the controls is one of the major tasks of the simulation.

There are two kinds of control units in the system: continuous controls and discrete controls.

Besides already available controllers in CRUISE, additionally, the development and parametrization of new controllers can be done with Matlab/Simulink. Additionally, Matlab/Simulink allows the generation of DLLs for coupled simulation and code generation for ECUs (see e.g. the homepage of Matlab/Simulink and code generation with realtime toolbox.<sup>1</sup>)

**Continuous** control units can be either a control which depends directly on input values, e.g.  $z = h(x, v, a)$  or which can be described by a differential equation  $\dot{z} = f(x, v, a, z)$ , e.g. from a PI control.

Besides continuous states, there are also **discrete** variables, states and control units in the model of the vehicle, e.g. gear or slip-stick condition. Changes in a discrete variable causes a state event and a change in the system matrix  $S$ . Controls which change the discrete variables can be described by inequality constraints of the form

$$g(d, x, v, a) \leq 0. \quad (5)$$

Additionally, other changes in discrete variables, e.g. from slip-stick, can be described by inequality constraints, too. For more on this hybrid systems see [18, 23] for more general approaches or [25] for specialized approach in vehicle dynamics.

## 2.3 Electrical System

Similar as for the mechanical system, also the electrical system in the vehicle is specialized. Components modelled are battery, generator and electrical engine, each with a detailed physical model (see [2]). There is no inductivity in the system and all hybrid parts are connected in parallel.

In our situation, the electrical components calculate their current  $i_j$  depending on the voltage level  $u$  in the subsystem. Taking now the voltage as additional variable, this leads to the additional equation for the electrical system per net

$$\sum_j i_j(u) = 0 \quad (6)$$

in which the sum goes over all electrical components in the net.

---

<sup>1</sup> MathWorks homepage <http://www.mathworks.com>.

*Remark.* Due to the specific models, the electrical system consists of resistors and capacities. Hence differentiation of (6) gives terms of the form  $R \cdot \dot{u}$  and  $C \cdot u$  with resistances  $R$  and capacities  $C$  which may depend on mechanical state,  $i, u$ , and additional system states like temperature. The once derived equation can be brought into the form  $\dot{u} = \text{right hand side}$ . Such that we have an index of one.

## 2.4 Coupled System

Putting all equations together, we get a coupled system with multibody part (second order ODE), control equations (first order part) and constraints (equality and inequality). Additionally, the variables in the system are either continuous or discrete. For calculating a stable solution, it is assumed that the discrete variables change their values only if an inequality constraint becomes active. Trivial connections can be substituted to reduce the number of unknowns,  $Ca = 0$  can be used to reduce the integration to the independent DOFs.

$$\begin{pmatrix} M & B \\ C & D \end{pmatrix} \cdot \begin{pmatrix} \ddot{x} \\ \dot{m} \end{pmatrix} = \begin{pmatrix} 0 \\ F(x, \dot{x}, u, z, d, i) \end{pmatrix} \quad (7)$$

$$\dot{z} = f(x, \dot{x}, u, z, d, i) \quad (8)$$

$$0 \geq g(x, \dot{x}, u, z, d, i) \quad (9)$$

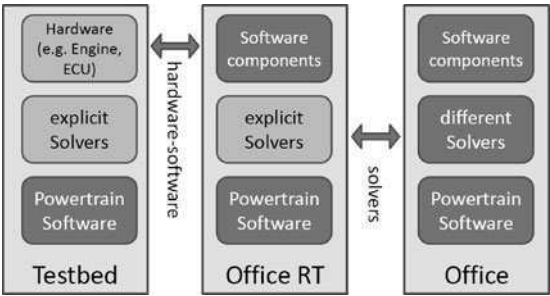
$$\sum_j i_j(x, \dot{x}, u) = 0 \quad (10)$$

## 2.5 Analysis Tasks

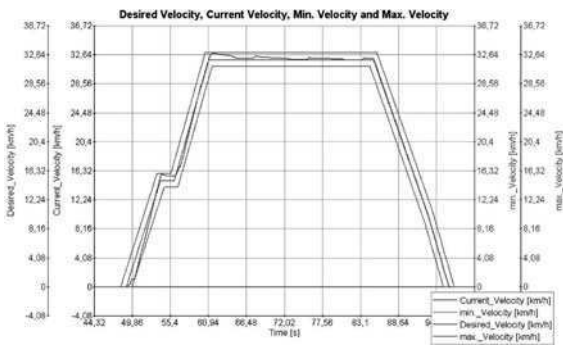
The focus is on the simulation and time integration of the system (7)–(10). Additionally needed boundary conditions, like initial values, environment values are given by the task simulation. The same vehicle equation system (7)–(10) is utilized on different platform (office, office RT, realtime/tesbed). This allows on the specific platform a detailed analysis, development and validation and verification of the calculation. Data can be collected on the realtime platform and used for parametrization on the office system. On the office system additional stationary analysis are possible for a more in depth analysis of the vehicle with additionally stationary tasks like constant driving conditions, inclination abilities.

Using the same physical model on the different platforms allows a detailed analysis of specific effects on the platforms (see Fig. 8). It is possible to test if effects occur due to the change from implicit to explicit time integration, replacement of software model by hardware (ECU controller), if the parametrization is accurate

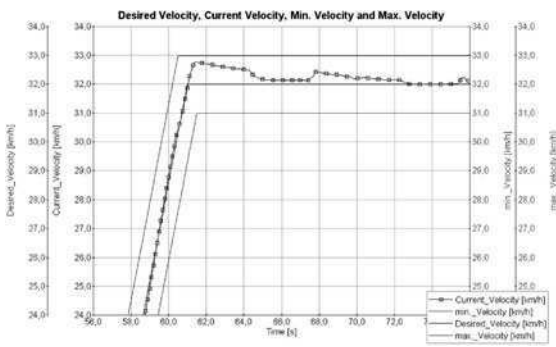




**Fig. 8** Differences between realtime/testbed (on the *left*), office realtime (*middle*) and office (*right*) situation



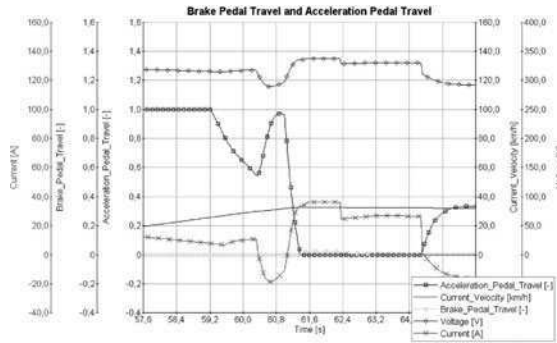
**Fig. 9** Clip from an office driving cycle simulation (ECE). Plotted are desired velocity, actual velocity and limits



**Fig. 10** Enlarged section from 9 with marked actual velocity

enough and similar. By this, a smooth transition on the realtime system is possible from software and software-in-the-loop models to hardware-in-the-loop.

In Figs. 9 and 10 a typical standard driving cycle simulation is depicted with a desired velocity and limits on the velocity of the vehicle. The driver tries to follow



**Fig. 11** Corresponding pedal movements for the ECE clip of Fig. 9

the desired velocity. The driver is a control unit which compares the desired and the actual velocity and calculates from this the pedal movements, see Fig. 11. Due to the dynamic of the car and the calculated pedal movements the specific fuel and electrical consumptions are calculated. In contrast to the fuel, the electrical energy can be reclaimed by the generator (with losses).

### 3 Time Integration

For the simulation the system is integrated with respect to time. In the off-line situation the total time interval is normally given beforehand, in the realtime case the simulation end time normally is open. In both cases the simulation should be accurate and efficient.

#### 3.1 Office Simulation

In the office or off-line situation, efficiency is reached by targeting large time step and using an implicit or semi-implicit integration step (implicit Euler step or Burlisch-Stoer method, see [3, 17]). But still, switches and state events in the system should be calculated correctly, otherwise oscillations may be triggered. To achieve this, an a priori step size adaptation [26] is used in which the state events are described by (the activation of) inequality constraints.

For the normal simulation the step size length in the range of 10 ms to 100 ms.

#### 3.2 Realtime Office

For development and analysis of problems, e.g. from the realtime simulation it is also possible to simulate the situation on the testbed system under realtime

conditions on the office computer. By this, one can easier detect problem e.g. due to the smaller time steps and parametrization, if an effect/oscillation is mainly related to the time step.

Additionally, in the development step one can easier profile the model and detect, e.g. evaluation limits, problems, by which parts performance is lost, and for example, which model complexity cannot be calculated in realtime.

### 3.3 Realtime

In contrast to the office simulation, the most important condition for realtime is that each time step must be finished within the given time frame. The normal time step length is 1 ms for resolving effects upto 500 Hz (limit from the NyquistShannon sampling theorem, see [14], Chap. 8). As noted (see Sect. 1), the numerical integration step should be finished within 0.5 ms CPU time.

Due to the sampling and communication with external parts, fixed time steps have to be used. Inner subdivisions of the interval, as for Runge-Kutta steps are possible, but the values from external parts are then extrapolated.

To achieve an efficient time step, as much as possible is put into the setup and initialization. Additionally, profiling of the model evaluation on the realtime office system is employed to identify time consuming code parts and finding more efficient linear algebra implementations additional assumption that  $\delta t = 1$  ms.

As timing experiments reveals, one has about time for 10 to 20 system evaluation (see Table 2 in Sect. 4.2). It is known, that even for fast automatic differentiation one has the limit of five times the system calculation effort for evaluating the sensitivity independent of the number of unknowns [12]. Which would leave to few iterations for nonlinear system solver. Hence we are restricted to explicit integrators.

For the system  $y' = f(t, y)$  we used (see [16, 17])

- Heun

$$k_1 = f(t_n, y_n) \quad (11)$$

$$y_{n+1} = y_n + \frac{h_n}{2} (f(t_n, y_n) + f(t_n + h_n, y_n + h_n k_1)) \quad (12)$$

and

- Bogacki-Shapine

$$k_1 = f(t_n, y_n) \quad (13)$$

$$k_2 = f\left(t_n + \frac{1}{2}h_n, y_n + \frac{1}{2}h_n k_1\right) \quad (14)$$

$$k_3 = f\left(t_n + \frac{3}{4}h_n, y_n + \frac{3}{4}h_n k_2\right) \quad (15)$$

$$y_{n+1} = y_n + \frac{2}{9}h_n k_1 + \frac{1}{3}h_n k_2 + \frac{4}{9}h_n k_3 \quad (16)$$

$$k_4 = f(t_n, y_{n+1}) \quad (17)$$

$$z_{n+1} = y_n + \frac{7}{24}h_n k_1 + \frac{1}{4}h_n k_2 + \frac{1}{3}h_n k_3 + \frac{1}{8}h_n k_4. \quad (18)$$

The Heun method is of order  $h^2$  with two function evaluations where as Bogacki-Shapine is of order  $h^3$  with four evaluations. The additional information about the error estimator from  $z_{n+1}$  in (13)–(18) is only used for additional internal subdivision of the interval, if possible. In case of Heun, the value  $h_n \|k_1\|$  can be used to estimate if subdivisions would be advantageous.

If the model allows, the usage of Bogacki-Shapine is advantageous due to the better order and additionally, if internal subdivisions are done, then  $k_4$  corresponds to  $k_1$  of the next step, such that only three additional function evaluations are needed. Due to possible switches and changes in the system at the beginning of the time step, the value of the former step cannot be re-used.

If a switch occurred at the beginning of the time step, no subdivisions is possible due to the additional work for the switch. If no switch occurred, at least one more may be possible (doing two  $\frac{h}{2}$  steps internally for the powertrain system). So in the internal subdivision control, one has to combine a time estimate with some stability or accuracy criteria from the formulas. If it is possible to estimate the time of the function evaluation  $f$  then additionally this can be used to determine the number of possible subdivisions.

Due to the algebraic constraint from the electrical equilibrium condition in (7)–(10) a purely explicit time integration is not sufficient for a stable and accurate solution [4]. In our case, no inductors are present in the electrical system, but only resistance and capacities. Hence the index of the constraint is 1 (with respect of having the voltage  $u$  as algebraic variable). Additionally, the electrical system reacts with a much higher speed than the mechanical system. Hence, in decoupling one can first solve for the mechanical system and then for the electrical equilibrium  $\sum i_j(x, v, u) = 0$  with fixed mechanical state  $x$  and  $v$ . As this reduces to a nonlinear, one-dimensional equation, the solution can be done efficiently by a few step of Regula-falsi.

Additionally, in evaluating the electrical components, the evaluation is reordered such that only the electrical parts, (10), needed for the constraint are evaluated. In doing so, the cost of the inversion of (10) for  $u$  is further reduced and the explicit solver sees only the ODE.

This time integration step can be seen as a mixed discretization in which the differential variables are discretized explicitly and the algebraic variables (in our case the voltage  $u$ ) is discretized implicitly.

## 4 Application

In this section we illustrate our approach with a typical industrial example.

## 4.1 Model

Besides the standard components like engine, tyres, brakes in the hybrid car the model includes additionally a battery, a generator and an electrical engine. The control of the engine and generator is given by a Matlab/Simulink controller which is transformed into a DLL for the off-line case and transferred to the hardware for the realtime case. The communication from the powertrain model to the controller goes via the signal bus similar as in the vehicle. Each system does not know whether the other is in software or in hardware. For the realtime case the vehicle, tyres are simulated by the testbed system.

In Fig. 12 the office model is depicted on the left and the realtime model on the right. The powertrain system is the same for both. In the realtime case the simulation of the vehicle, tyres and cockpit is done by other parts of the testbed system. Moments and rotational state at these points are transferred between powertrain and testbed system.

In the vehicle, an engine, an electrical motor and a generator are connected via a planetary gear to the output driveline. For the overall performance of the vehicle, the interplay of the mechanical and hybrid components is essential. To achieve this, the different controls have to be well adjusted for the performance and fuel consumption, e.g. with energy recuperation.

## 4.2 Simulation Results

In this section we compare the realtime office calculation with the office calculation with respect to numerical performance. As Windows XP is not a realtime OS, there are some measure errors and peaks due to OS events. But for the overall performance the calculations show the same characteristics and performance as on the realtime testbed systems. Additional analysis can be included on the office system which are not possible on the testbed.

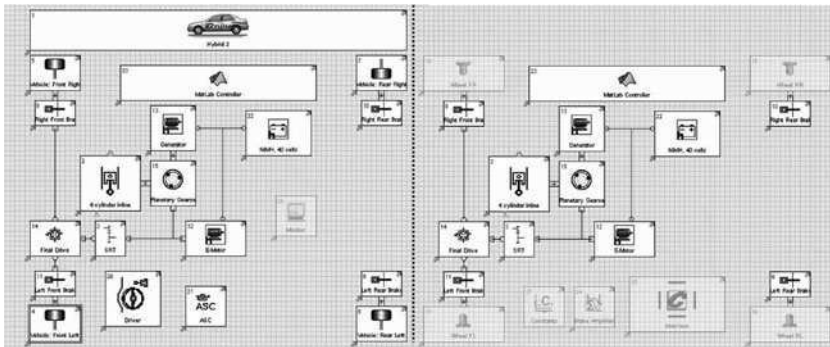


Fig. 12 Model for Office System (*left*) and Realtime System (*right*)

4.2.1 Realtime Calculations

In the realtime system the powertrain is calculated within our part. The calculation of the pedal movements, the load, the vehicle manoeuvre is done in the testbed system.

During the driving of a test case, the control splits the load signal as given by the driver (here calculated by the driver component) into load signals for the different engines and generator. Calculated curves are plotted in Fig. 13.

In Fig. 14 a clip of the timing is depicted (with state curves of the system). Critical is that it is possible to keep here the total CPU usage below 0.5 most of the time, though one can see some increase here due to gear changes and driver actions from the low average value of 0.2.

Calculating offline, it is possible to track the number of iterations needed for the embedded solution of the constraint. An increase in the number of steps in Fig. 15 corresponds to fast changes in the electrical systems and results in higher overall

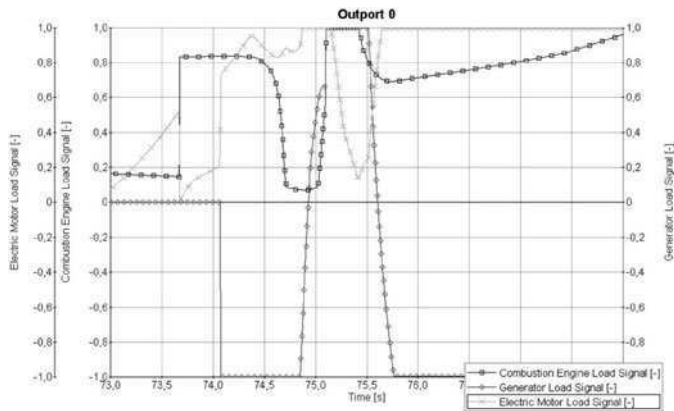


Fig. 13 Load signals from the control Unit to the different engines and generators

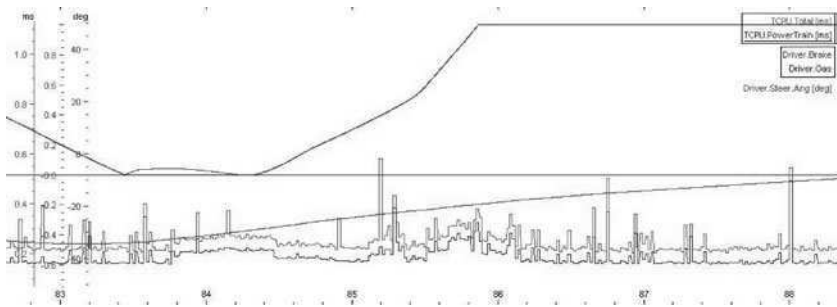
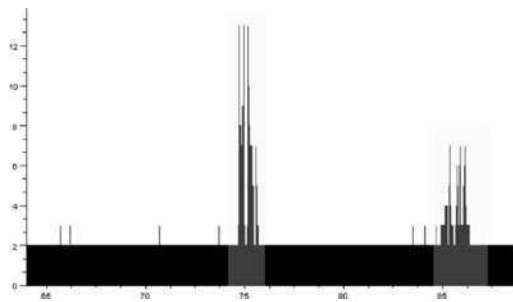


Fig. 14 Timing for RT Office Calculation with additional state curves. Timing curves of the powertrain and the full system are plotted in the lower part. The total is above the timing for the powertrain. The straight line in the middle corresponds approximately the 0.5 line

**Fig. 15** Number of nonlinear solution steps for the constraint over time. Around  $t = 75$  and  $t = 86$  s, there are fast changes of the electrical system



CPU times for this time step. Nevertheless, as the iteration is done only for a subset of the system, the overall performance can be kept within limits. One can see that a higher number of iterations in Fig. 15 corresponds to changes in the system (visible in the load signals in Fig. 13) which implies higher changes. But as one can see in Fig. 14, the additional CPU demand is small.

#### 4.2.2 Comparison Implicit and Explicit Solution

For estimating the accuracy of the testbed calculation, we compared the calculation with an implicit solver with 1 ms time step on the realtime office system. For the nonlinear system solution we used an Quasi-Newton method in which the Jacobian is kept constant, but recalculated if accuracy and performance criteria fail. The needed CPU time for the nonlinear solution is in average double the time as for the explicit case. This would be still sufficient for the constraints for the testbed system, giving an overall performance of about 0.4 to 0.6. But due to the nonlinear system solution, ever 10th or 20th time step (due to Jacobian calculations) the realtime constraint is violated. Although this does not lead to an extreme increase in the average CPU time demand, this makes the implicit method only useful for comparison in the realtime office case.

In Fig. 16 we plotted the voltage and current curves for the battery for the implicit and explicit solution. In calculating the velocity and the way of the vehicle, we have a difference in the way of 0.14% after 100 s simulation due to numerical drift. As the controls normally depend on the way (e.g. curve, acceleration possible, ...) we have plotted the curves in Fig. 16 over the way. Depending on time, we would have a shift of approximately 0.2 s.

#### 4.2.3 Comparison Realtime and Office

For comparison, the office driving cycle of the Fig. 9 with 195 s simulated time is calculated in 3.23 s total time. The configured step size length is 10 ms. The minimum step length from the step size control was 2 ms and the average 9.8 ms. Hence for most of the time steps the maximum step length was possible.

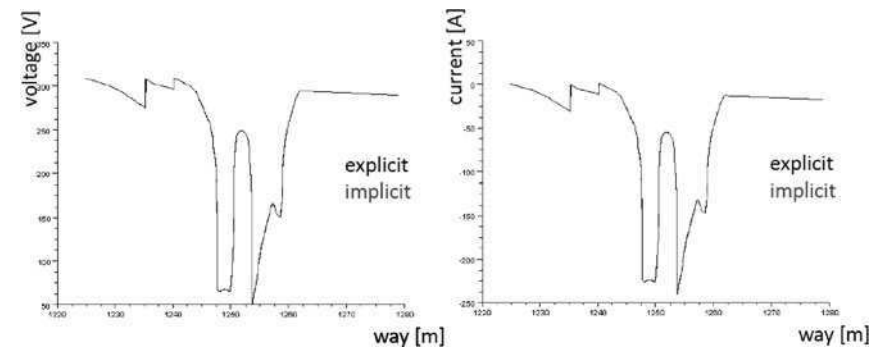


Fig. 16 Battery voltages explicit and implicit over way

**Table 1** Number of function evaluations per time step of either 1 ms or 10 ms length. Different real-time (rt) integrators compared to office simulation (with ECE test cycle). BogSha = Bogachi-Shampine, B.S. = Burlisch-Stoer

	Average	Min	Max
rt Heun 1 ms	9.99	6	10
rt BogSha 1 ms	8	8	8
rt Impl.Euler 1 ms	16	4	24
office B.S. 10 ms	23.6	16	1,265
office Impl. Euler 10 ms	8.4	4	145
office Impl. Euler 1 ms	5.7	4	142

**Table 2** Estimated time per system evaluation

Integrator	Estimate
rt Heun	1.70238e-002 ms
rt BogSha	1.7542e-002 ms
office Heun	1.70477e-002 ms
office BogSha	1.72632e-002 ms

4.2.4 Comparison with Respect to System Evaluation

In the office RT environment it is possible to include some additional timers and counters to estimate the time needed per system evaluation and the number of evaluations per time step. The measurements for the example are given in the tables (1) and (2).

Due to the complexity of the system evaluation for counting the number of evaluations in table (1) the number of evaluations of the engine component has been monitored. For the timing estimate the time in the integrator routines (with all overhead) has been measured. Due to the few more operations for Bogachi-Shampine integrator the estimate is slightly higher than for Heun. The time for a refresh is included, but is not seen that much in the average.



From profiling it is known, that the time needed for the update of system matrices after an state event is about 0.1 to 0.2 ms. Additionally, the calculation of the other parts of the testbed system need about 0.2 ms. Putting this together with the tables (1) and (2) gives a good agreement with Fig. 14.

Also one can estimate, an iterative solver would fail the realtime time constraint (with upto 24 evaluations per time step).

## 5 Conclusion

From the model and the mathematical simulation requirements it is a larger step to go from offline simulation to real-time. By analysing the model and the numerical algorithm, it is possible to simulate extended drivelines under real-time conditions with 1 ms step size as needed for realistic application. This allows a smooth transition with respect to platform, but also with respect to software-in-the-loop model and hardware-in-the-loop model which is advantageous in the engineering development process allowing a smooth transition from office to testbed calculations and an integrated approach in the development for testing and verification.

**Acknowledgements** The work of Ralf Pfau is supported by the “Bundesministerium für Wirtschaft und Arbeit” and by the government of Upper Austria within the framework “Industrielle Kompetenzzentren und Netzwerke”.

## References

1. Arnold M, Burgermeister B, Eichberger A (2007) Linearly implicit time integration methods in real-time applications: DAEs and stiff ODEs. *Multibody Syst Dyn* 17:99–117
2. AVL Cruise. Theory Manual and User Guide, Version 3.0 (January 2005) AVL, Graz
3. Bader G, Deuffhard P (1983) A Semi-implicit mid-point rule for stiff systems of ordinary differential equations. *Numer Math* 41:373–398
4. Brenan KE, Campbell SL, Petzold LR (1996) Numerical solution of initial-value problems in differential-algebraic equations. SIAM, Philadelphia, PA
5. Burgermeister B, Arnold M, Eichberger A (2009) Smooth velocity approximation for constrained systems in real-time simulation. In: Proceedings of multibody dynamics 2009, ECCOMAS thematic conference. Warsaw
6. Burgermeister B, Arnold M, Esterl B (2006) DAE time integration for real-time applications in multi-body dynamics. *Z. Angew. Math. Mech.* 86(10):759–771
7. Cuadrado J, Dopico D, Naya MA, Gonzalez M (2004) Penalty, semi-recursive and hybrid methods for MBS real-time dynamics in the context of structural integrators. *Multibody Syst Dyn* 12:117–132
8. Elmqvist H, Mattsson S, Olsson H (2002) New methods for HIL simulation of stiff models. In: Proceedings of 2nd international modelica conference. Oberpfaffenhofen, Germany
9. Elmqvist H, Mattsson S, Olsson H, Andreasson J, Otter M, Schweiger C, Brück D (2003) Real-time simulation of detailed automotive models. In: Proceedings of 3th international modelica conference. Linköping, Sweden
10. Elmqvist H, Mattsson S, Olsson H, Andreasson J, Otter M, Schweiger C, Brück D (2004) Real-time simulation of detailed vehicle and powertrain dynamics. In: Electronics simulation

- and optimization (SAE 2004 World Congress). Detroit, USA, March, 2004, SAE International. Document Number: 2004-01-0768
11. Esterl B, Butz T, Simeon B, Burgermeister B (2007) Real-time capable vehicle-trailer coupling by algorithms for differential-algebraic equations. *Vehicle Syst Dyn* 45:819–834
  12. Fischer HC (1990) Schnelle automatische Differentiation, Einschliessungsmethoden und Anwendungen. PhD Thesis, University Karlsruhe, Germany
  13. Fischer R, Kirsten K (2006) The turbo-hybrid: Holistic approach for a modern gasoline hybrid drive. In: Proceedings of 27th international Vienna motor symposium. Vienna, Austria
  14. Föllinger O (1992) Regelungstechnik. Hüthig, Heidelberg
  15. Gonzalez M, González F, Dopico D, Luaces A (2007) On the effect of linear algebra implementations in real-time multibody system dynamics. In: Bottasso C, Masarati P, Trainelli L (ed) *Multibody dynamics*, 2005, Milano, Italy, pp 25–28
  16. Hairer E, Nørsett SP, Wanner G (2000) Solving ordinary differential equations I: Nonstiff problems. Springer, Berlin
  17. Hairer E, Wanner G (2002) Solving ordinary differential equations: Stiff and differential-algebraic problems. Springer, Berlin
  18. Hamann P, Mehrmann V (2005) Numerical solution of hybrid differential-algebraic equations - Available via Matheon [http://www.matheon.net/research/list\\_preprints.asp](http://www.matheon.net/research/list_preprints.asp)
  19. Hasewend W (1998) Verfahren zur Berechnungsgestützten Entwicklung moderner Antriebssysteme. PhD Thesis, Technical University Graz, Austria
  20. de Jalón JG, Bayo E (1994) Kinematic and dynamic simulation of multibody systems – the real-time challenge. Springer, New York
  21. Kim S, Jeong WH, Tak TO, Kim LK (2009) Performance evaluation of intelligent chassis controller using HIL-simulator based on real-time multibody vehicle dynamics. In: Proceedings of multibody dynamics 2009, ECCOMAS Thematic Conference. Warsaw
  22. Korkealaakso PM, Rouvinen AJ, Moisio SM, Peusaari JK (2007) Development of a real-time simulation environment. *Multibody Syst Dyn* 17:177–194
  23. Mehrmann V, Wunderlich L (2008) Hybrid systems of differential-algebraic equations Analysis and numerical solution. Available via Matheon [http://www.matheon.net/research/list\\_preprints.asp](http://www.matheon.net/research/list_preprints.asp)
  24. Morawietz L, Risse S, Christ T, Zellbeck H, Reuss H (2005) Modelling an automotive power train and electrical power supply for HiL applications using Modelica. In: Proceedings of 4th international modelica conference. Hamburg University of Technology, Hamburg-Harburg
  25. Pfau RU (2003) Numerical algorithms for the simulation and optimization of vehicle driving performance and fuel consumption. PhD Thesis, University Linz, Austria
  26. Pfau RU (2007) A priori step size adaptation for the simulation of nonsmooth systems. *Commun Numer Meth Eng* 23(2):85–96
  27. Pfau R, Schaden T (2009) Offline and real-time simulation of vehicle drivetrain dynamics. In: Proceedings of SIAM conference on computational science and engineering (CSE09). Miami, USA
  28. Ramaswamy D, McGee R, Sivashankar S, Deshpande A, Allen J, Rzemien K, Stuart W (2004) A case study in hardware-in-the-loop testing: Development of an ECU for a hybrid electric vehicle. SAE Technical Paper Series
  29. Rill G, Chucholowski C (2007) Real time simulation of large vehicle systems. In: Proceedings of multibody dynamics 2007, ECCOMAS thematic conference. Milano, Italy
  30. Rulka W, Pankiewicz E (2005) MBS approach to generate equations of motions for HIL-simulations in vehicle dynamics. *Multibody Syst Dyn* 14:367–386
  31. Schiela A, Bornemann F (2003) Sparsing in real time simulation. *Z. Angew. Math. Mech.* 83:637–647
  32. Schiela A, Olsson H (2000) Mixed-mode integration for real-time simulation. In: Proceedings of Modelica Workshop 2000, Lund, Sweden, pp 69–75
  33. Schlegel C, Bross M, Beater P (2002) HIL-Simulation of the Hydraulics and Mechanics of an Automatic Gearbox. In: Proceedings of 2nd international modelica conference. Oberpfaffenhofen, Germany

34. Schuette H, Waeltermann P (2005) Hardware-in-the-loop testing of vehicle dynamics controllers a technical survey. In: Proceedings of 2005 SAE world congress. Detroit, Michigan
35. Soejima S, Matsuba T (2002) Application of mixed mode integration and new implicit inline integration at Toyota. In: Proceedings of 2nd International modelica conference, pp 65-1–65-6. Oberpfaffenhofen, Germany
36. Urch P, Huss A, Ebner P, Jürgens G, Pfau R (2009) Integrierte Fahrzeug Systemsimulation im Entwicklungsprozess von früher Konzeptphase bis zu Testläufen. In: Proceedings of 2. Grazer Symposium “VIRTUELLES FAHRZEUG”. Graz, Austria, April 27–28, 2009
37. Willumeit HP (1998) Modelle und Modellierungsverfahren in der Fahrzeugdynamik. B.G. Teubner, Stuttgart, Leipzig
38. Wu Y, Chen B, Hsieh F, Huang M, Wu Y (2006) Development of hardware-in-the-loop simulation for scooter engine control. SAE Technical Paper Series
39. Zomotor A (1991) Fahrwerktechnik: Fahrverhalten. Vogel Fachbuch: Kraftfahrzeugtechnik. Würzburg

# Assessment of Antagonistic Muscle Forces During Forearm Flexion/Extension

Maxime Raison, Christine Detrembleur, Paul Fisette, and Jean-Claude Samin

**Abstract** Today, the accurate assessment of muscle forces performed by the human body in motion is still expected for many clinical applications and studies. However, as most of the joints are overactuated by several muscles, any non-invasive muscle force quantification needs to solve a redundancy problem. Consequently, the aim of this study is to propose a non-invasive method to assess muscle forces in the human body during motion, using a multibody model-based optimization process that attempts to solve the agonistic and antagonistic muscle overactuation. The main originality of the proposed method is the cautious using of Electromyographic (EMG) data information, known by all to be noisy-corrupted, via a protocol divided into two main steps:

1. Muscle force static calibration.
2. Muscle force dynamical quantification.

In this chapter, the process is applied to a benchmark case: the force quantification of the elbow flexor and extensor muscle sets of subjects engaged in weightlifting and performing cycles of forearm flexion/extension. A statistical validation of this method shows a good inter-test reproducibility and a very good correlation between a. the net joint torques resulting from the obtained muscle forces and b. the net joint torques given by inverse dynamics.

Consequently, since the method is able to consider measured information on the actual muscle activation, it becomes a promising alternative to methods based on preset strategies, usually presented in literature, such as the strategy that maximizes endurance defined by Crowninshield et al.

---

M. Raison (✉), P. Fisette, and J.-C. Samin  
Center for Research in Mechatronics (CEREM), École Polytechnique de Louvain – Université catholique de Louvain, Bâtiment Stévin, Place du Levant 2, 1348 Louvain-la-Neuve, Belgium  
e-mail: [maxime.raison@uclouvain.be](mailto:maxime.raison@uclouvain.be); [paul.fisette@uclouvain.be](mailto:paul.fisette@uclouvain.be); [jc.samin@uclouvain.be](mailto:jc.samin@uclouvain.be)  
<http://www.cerem.be>

C. Detrembleur  
Rehabilitation and Physical Medicine Unit (READ), Université catholique de Louvain,  
Tour Pasteur, Avenue Mounier 53, 1200 Bruxelles, Belgium  
e-mail: [christine.detrembleur@uclouvain.be](mailto:christine.detrembleur@uclouvain.be) <http://www.uclouvain.be/en-read.html>

# 1 Introduction

Today, the accurate assessment of internal efforts, and particularly the muscle forces performed by the human body in motion, is still expected for applications in many fields, like:

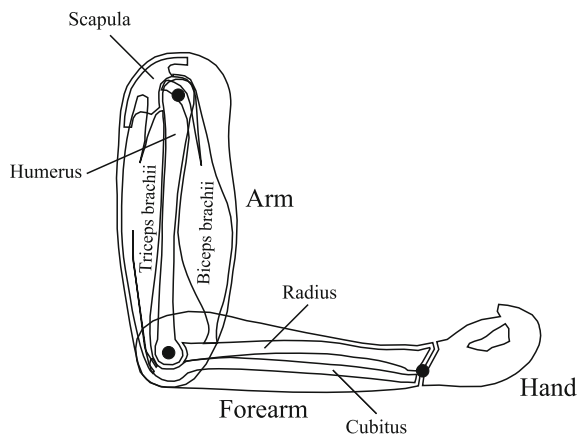
- Rehabilitation: for the evaluation and follow-up of patients with musculo-skeletal pathologies, e.g. using gait analysis for hemiparetic [16] or scoliotic [30] patients.
- Ergonomics: for instance, for comfort analysis of vehicle drivers [28, 37] or during vehicle accessibility motion [14].
- Prevention: in order to avoid the risks of wounds and the appearance of pathologies associated with motions, e.g. during maximal pushing efforts [6, 33].
- Sports: in order to analyse and improve athletic performances, e.g. during pedalling [22] or somersault on trampoline [4].

However, as most of the joints are overactuated by several muscles (2.6 muscles in average per *degree of freedom* (DOF) in the human body [1]), any non-invasive muscle force quantification needs to solve a redundancy problem.

## 1.1 Redundancy Problem Formulation

Let us consider, for instance, the flexion/extension of the human elbow. As represented in Fig. 1, this system is composed of three body members, the arm, the forearm and the hand, which are articulated around joints, the shoulder, the elbow and the wrist, respectively. Particularly, the elbow joint flexion/extension is actuated by:

- The *flexor* (or *agonistic*) muscle set, mainly composed of the longus and brevis biceps brachii.



**Fig. 1** Illustration of the human elbow, including a representation of the flexor muscle sets (biceps brachii) and extensor muscle sets (triceps brachii) that overactuate the elbow joint

- The *extensor* (or *antagonistic*) muscle set, mainly composed of the longus and lateral triceps brachii.

Consequently in this example, we can state that on the one hand, pure flexion/extension corresponds to one rotational DOF of the joint elbow, and that on the other hand, there is one flexor muscle set and one extensor muscle set, which together overactuate the elbow joint.

In a dynamical context, the link between the individual muscle forces produced about the joint and the resulting moments can be also stated via the *Principle of Potential Power*, as follows:

$$\mathbf{M}_E \cdot \Delta\boldsymbol{\omega}_E = \sum_{i=1}^N \mathbf{F}_{m,i} \cdot \Delta\mathbf{v}_i \quad (1)$$

where:

- $\mathbf{M}_E$  is the resultant moment at the joint, i.e. at the elbow, in our example of flexion/extension.
- $\mathbf{F}_{m,i}$  is the force vector performed by the  $i$ th muscle set, for  $i = 1, \dots, N$  muscle sets; here this corresponds to the flexor set ( $i = 1$ ) and extensor set ( $i = 2$ ).
- $\Delta\boldsymbol{\omega}_E$  is the potential variation of the angular velocity vector at the elbow during flexion/extension.
- $\Delta\mathbf{v}_i$  is the potential variation of the translation velocity vector of the  $i$ th muscle set insertion points, for  $i = 1, \dots, N$  muscle sets; here this corresponds to the flexor set ( $i = 1$ ) and extensor set ( $i = 2$ ); both  $\Delta\boldsymbol{\omega}_E$  and  $\Delta\mathbf{v}_i$  must be chosen compatible with the kinematic description of the motion [42].

Practically, as vectors  $\Delta\boldsymbol{\omega}_E$  and  $\Delta\mathbf{v}_i$  can be determined using kinematic measurements and anatomical information, and as the joint resultant moment  $\mathbf{T}_E$  can be determined using inverse dynamics, the only unknown variables in (1) are the muscle forces,  $\mathbf{F}_{m,i}$ . As the motion of forearm pure flexion/extension is here considered as having one DOF, equation (1) is a mathematically redundant system that has an infinite number of possible solutions for the muscle force repartition between  $\mathbf{F}_{m,1}$  and  $\mathbf{F}_{m,2}$ .

## 1.2 Classification of Solving Methods

Among the methods to solve the undetermined problem of the muscle redundancy, most frequently used methods have been based on optimization function, which attempted to model the muscle co-contraction strategy during motion. Considering the cost function, many ones have been proposed in literature, on the basis of criteria that minimized the metabolic cost [41], the sum of forces [35], a weighted sum of forces [27], or the fatigue [20]. Particularly, a significant advance in the theoretical prediction of individual muscle forces was made by Crowninshield and Brand [13], who insisted on the importance of an objective function based on

experimental tests (rather than an arbitrary or mathematically convenient one). These investigators used a nonlinear optimal design formulation. The objective function was based on the idea that during many cyclic, activities with low-level control (such as normal walking during steady state), muscles are recruited in such a way as to *maximize endurance* (i.e. maximize the duration a motion can be sustained). Concretely, they formalized the endurance maximization by minimizing the following objective function [13]:

$$\phi = \sqrt[n]{\sum_{i=1}^N \left( \frac{F_{m,i}}{PCSA_i} \right)^n} \quad (2)$$

where:

- $F_{m,i}$  [N] = force magnitude of the  $i$ th muscle.
- $PCSA_i$  [m<sup>2</sup>] = physiological cross-sectional area of the  $i$ th muscle.
- $N$  [-] = total number of muscles considered.
- $n$  [-] = power, usually set equal to 2 or 3 in order to best fit the experimentally measured patterns of forces using electromyographic data [21].

The associated inequality constraints are:

$$F_m^i \geq 0, \text{ for } i = 1, \dots, N \quad (3)$$

and equality constraints are given by (1).

Today, most of the methods of muscle force quantification are strictly based on Crowninshield cost function, although these authors insisted on the fact that their method was simply an approximation. For instance, they showed that, during gait, the estimated muscle forces were only close to the EMG patterns during the swing phase, and were significantly different during the other phases. As conclusion of their discussion, the authors of this method attempt to insist on the importance of using muscle activity information in order to improve the quality of results obtained by any optimization process. Indeed, this reference points out the fact that the proposed methods provide more physiologically realistic muscle force results than the classical methods, because the results are more consistent with both EMG and physiological constraints under dynamical conditions. However, it is known that EMG is reasonably proportional to the muscle force in *static* conditions [5], while the “EMG - force” relation is not linear in dynamical conditions and has not yet been reliably established [21].

### 1.3 Objective of this Study

In this context, the aim of the present study is the development of a non-invasive method to quantify the muscle efforts of the human body in motion, on the basis of a model-based optimization that attempts to solve the muscle redundancy problem,

including a cautious use of EMG data. The process is applied to a benchmark case: the force quantification of the elbow flexor and extensor muscle sets of subjects engaged in weightlifting and performing cycles of forearm flexion/extension.

## 2 Material and Methods

This section will describe the principle of our method, the experimental set-up, the model and hypotheses, and finally the process of the muscle force quantification.

### 2.1 Principle

The principle of our method, schematically outlined by Fig. 2, is divided into two main steps: the muscle force calibration in statics and the muscle force quantification in dynamics.

#### 2.1.1 Protocol Step 1: Force Calibration

While EMG is reasonably proportional to the muscle force in *static* conditions [5], we must be aware that EMG is not sufficiently reproducible if the data recording does not exactly follow the same protocol. Consequently, we decided to calibrate the forces of the flexor and extensor muscle sets in isometric conditions, using proper EMG filtering and the Hill model [46]: here (Fig. 2, step 1), the elbow angle is  $90^\circ$  in the sagittal plane; the subject pulls on a rope fixed to a strain gauge on the floor (flexors activated), then the subject pulls on a rope fixed to this strain gauge via a pulley on the ceiling (flexors and extensors activated). This calibration gives the scale (factor  $K$ ) and offset (factor  $\Delta$ ) that will be used to calibrate muscle forces during the flexion/extension, which will only be used as “not too bad” initial values of forces as input for the muscle overactuation solving (Fig. 2, step 2).

#### 2.1.2 Protocol Step 2: Force Quantification

As soon as the calibration is completed, the subjects, still equipped with exactly the same optokinetic and EMG sensors, are engaged in weightlifting and perform several cycles of forearm flexion/extension (Fig. 2, step 2). For each of these trials, the model-based process comprises three consecutive steps, detailed in Sect. 2.4:

1. **Kinematics identification:** an optimization process that estimates the joint configurations,  $q_{mod}$ , of the MultiBody System (MBS) that best fits the experimental joint configurations,  $q_{exp}$ , measured by optokinetic sensors. The corresponding velocities,  $\dot{q}$ , and accelerations,  $\ddot{q}$ , are determined using numerical derivatives.



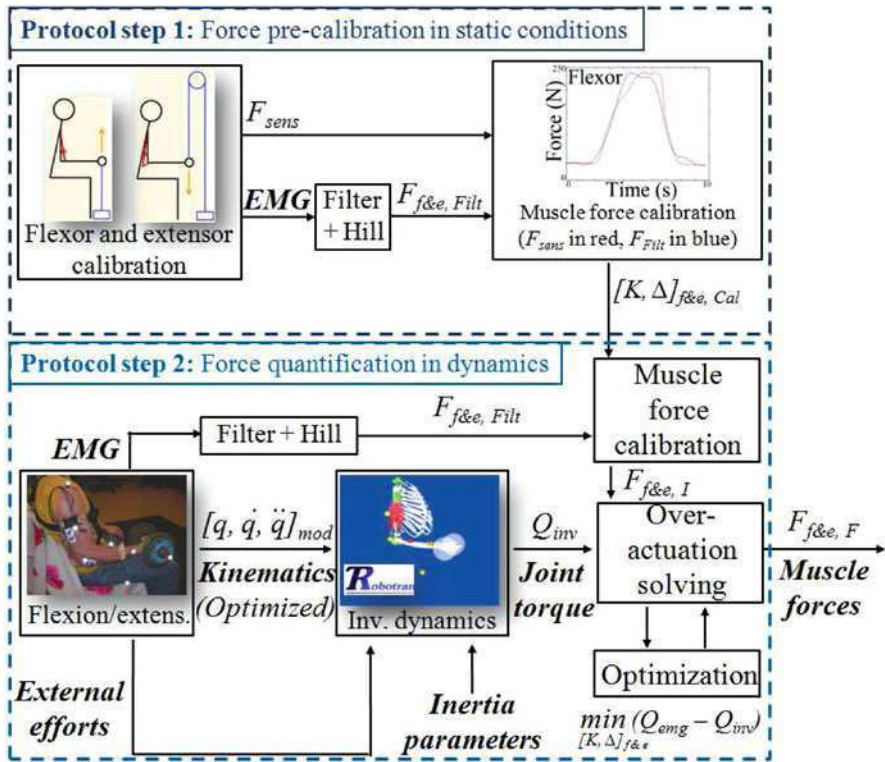
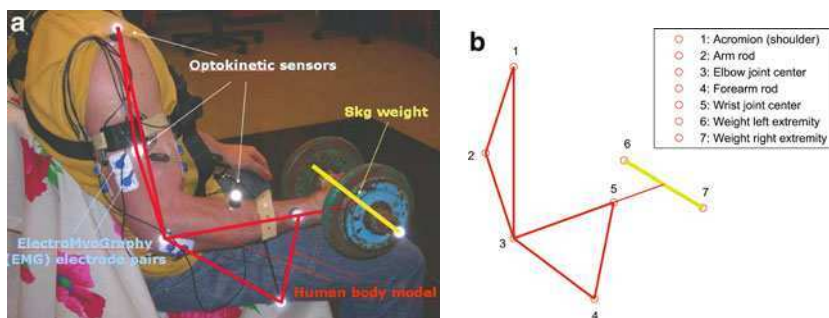


Fig. 2 General principle of the muscle force quantification, featuring both protocol steps

- Inverse dynamics:** an inverse dynamical model that provides the elbow contributive net torque  $Q_{inv}$  via recursive Newton–Euler equations of motion of the MBS, available in symbolic form [34].
- Muscle overactuation solving:** an optimization process that computes the forces of the flexor and extensor muscle sets during forearm flexion/extension. Starting from the  $K$  and  $\Delta$  factors given by step 1 of the protocol and the filtered flexor/extensor forces during flexion/extension, this optimization adapts  $K$  and  $\Delta$  so that the elbow net torque  $Q_{emg}$ , computed from the muscle forces, given their insertion points, best fits the elbow net torque  $Q_{inv}$  given by the inverse dynamics.

## 2.2 Experimental Set-up

The measurement set-up (Fig. 3a) is composed of seven optokinetic sensors (*Elite-BTS* acquisition system [7]) fixed at the subject's joint landmarks (in Fig. 3b: points



**Fig. 3** (a) Illustration of the subject carrying an 8 kg weight and performing several cycles of forearm flexion and extension, featuring the optokinetic sensors (reflecting spheres) and the EMG electrodes (double grips). Superimposition of the MBS (segments) on the basis of the weight configuration and the optokinetic sensors. (b) Description of the model, featuring the seven optokinetic sensors, based on anatomical landmark tables [18], that define the two articulated rigid bodies (the arm and the forearm), each defined by three points, articulated via two spherical joints (the acromion, representing the shoulder, and the elbow joint center)

from 1 to 5, plus both extremities of the weight axis), and also ElectroMyoGraphic (or EMG) data (*Elite-BTS*), which allow us to calculate the activation of the considered muscles during the motion.

The optokinetic data are sampled at 100 Hz and filtered by a 15 Hz adaptive low-pass numerical filter. The EMG data are first sampled at 1,000 Hz, then synchronized with the data at 100 Hz, after being rectified and filtered by a 5th order Butterworth low-pass filter [46].

The experiments were performed by healthy subjects related to our laboratory, who gave their informed consent to perform the experiments.

### 2.3 Model and Hypotheses

First of all, the system is modeled as a constrained MBS, using kinematic loops and cutting procedure (ball cuts), in order to take the muscle connection into account. As we are using the coordinate partitioning method, independent and dependent variables are systematically used to model this closed-loop system. In detail, the MBS (the segments in Fig. 3b) is composed of two rigid bodies (the arm and the forearm) articulated around spherical joints (the shoulder and the elbow), giving a total of six independent variables. A third body, the hand, is attached to the forearm via a locked revolute joint. Finally, both flexor (biceps brachii) and extensor (triceps brachii) muscle sets are also considered as bodies with mass and inertia parameters. The shoulder joint is considered as the reference point of the model, given that the subject is supposed to keep the shoulder fixed during the tests: practically, we checked that the subject did not move the shoulder very much during each trial and

we systematically subtracted the values of the shoulder kinematic components from all kinematic data in order to consider the shoulder as fixed at the inertial reference point (0, 0, 0) in our model.

A few characteristics and assumptions must be formulated about the different sets of inputs:

- The shoulder joint being considered as the reference point of the model, the forces  $\mathbf{F}_{ext}$  and torques  $\mathbf{M}_{ext}$  between the body and its environment, i.e. the “upper-shoulder”, are constraint forces that will not contribute to the equations of motion, and thus do not need to be measured. They can possibly be computed using the action-reaction principle.
- The body inertia parameters, i.e. the masses  $m_j$ , inertia moments  $I_j$  and centre of mass positions  $\overrightarrow{OM}_j$  of the  $j^{th}$  body member are taken from the tables of inertia from de Leva [18]. The inertia parameter identification is not part of this research: indeed, previous investigations [11] showed that non-invasive in-vivo dynamical identifications of body parameters are presently inappropriate for human body dynamics, because the resulting body parameters have significant inaccuracies due to experimental errors in the input data, related to the configuration of the bodies, and the external force and torque measurements.
- The system configuration, i.e. the experimental absolute coordinates  $x_{exp}(t)$  of the reference points, are measured by the six optokinetic sensors. The corresponding joint coordinates  $q$  of the MBS are numerically determined by a kinematic identification process and, after a low-pass filter, the corresponding velocities  $\dot{q}(t)$  and accelerations  $\ddot{q}$  are presently estimated from  $q$  by numerical differentiation techniques.

Considering the joint kinematics, we are aware that more adequate elbow and shoulder joint models could be used: in particular, previous studies [3] have developed more complex three-dimensional joints for the shoulder. The present model has been implemented with spherical joints, which is sufficiently accurate [31] at present with relation to our muscle overactuation solving method, but the model should be extended in the future in order to include more sophisticated joints. Finally, let us note that the elbow flexion/extension is physiologically defined using only one DOF. However, in practice, the axis orientation of the instantaneous relative velocity vector  $\boldsymbol{\Omega}_E$  at the elbow continuously changes during flexion/extension. Consequently, it is very important to state that, instead of considering only one DOF for the flexion/extension, it is more accurate to consider three DOFs (i.e. a spherical joint) at the elbow in order to take into account the variation of the  $\boldsymbol{\Omega}_E$  axis, and then to further project the elbow vector torque  $\mathbf{Q}$  onto the  $\boldsymbol{\Omega}_E$  axis, in order to obtain the elbow joint net torque  $Q_{inv}$  along the instantaneous rotation axis that contributes to the motion of flexion/extension, as follows:

$$Q_{inv} = \mathbf{Q} \cdot \frac{\boldsymbol{\Omega}_E}{\|\boldsymbol{\Omega}_E\|} \quad (4)$$

## 2.4 Process of Muscle Force Quantification

As introduced in Sect. 2.1, the protocol and the underlying calculation process are divided into two main steps: the muscle force calibration in statics (detailed in Sect. 2.4.1) and the muscle force quantification (Sect. 2.4.2).

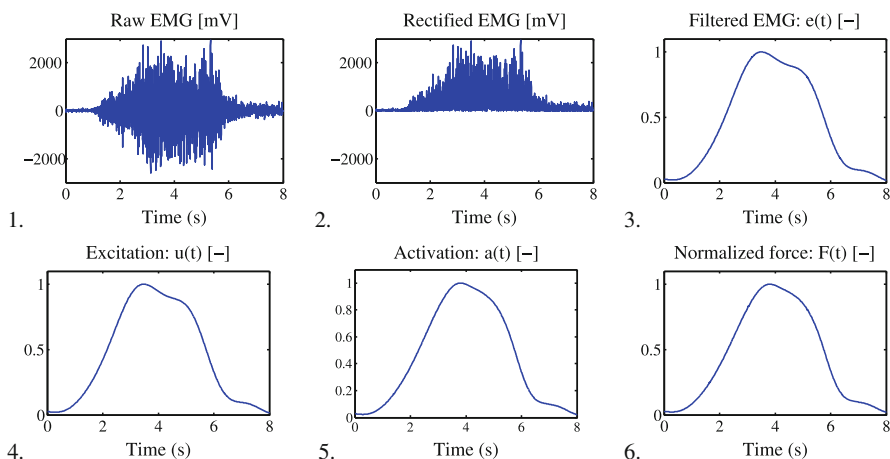
### 2.4.1 Muscle Force Calibration

While the EMG is reasonably proportional to the muscle force in *isometric* conditions [5], we must be aware that the EMG is not sufficiently reproducible if the data recording does not exactly follow the same protocol. Consequently, we decided to calibrate the forces of the flexor and extensor muscle sets in isometric conditions, using proper EMG filtering and the Hill model [46]: here (Fig. 2, step 1), the elbow angle is  $90^\circ$  in the sagittal plane; the subject pulls on a rope fixed to a strain gauge on the floor (flexors activated), then the subject pulls on a rope fixed to this strain gauge via a pulley on the ceiling (flexors and extensors activated).

EMG processing, in static or dynamical conditions, can be divided into five sequential steps to transform the raw EMG data of a muscle to its calibrated force, as illustrated in Fig. 4 and described in the following sections.

#### EMG Rectification

First of all, the raw EMG signal must be rectified, i.e. the absolute values of each signal sample are taken, in order to take into account the fact that the EMG electrodes record a signal of muscle fiber polarization–depolarization [46].



**Fig. 4** Illustration of the EMG Processing, representing the sequential transformations from the raw EMG data to the normalized muscle force  $F(t)$

## EMG Filtering

High-frequency noise, due to electrical interference such as wiring motion, must be removed from the signal by applying a low-pass filter that has zero-phase delay properties, so filtering does not shift the signal in time; in practice, we have chosen the commonly recommended forward and reverse low-pass 5th order Butterworth filter, with a cut-off frequency of 15 Hz in order to be sure to take the muscular contraction frequencies into account (knowing that the highest known muscular contraction frequencies are 12 Hz for a stressed muscle). Finally, this signal must be normalized with respect to its maximal value, in order to obtain a number between 0 and 1, representing the muscle fiber recruitment rate. Once this process is completed, the transformed EMG signal becomes the *filtered* signal labeled  $e(t)$ .

## Neural Activation

The muscle does not contract exactly at the time instant at which the motor unit is triggered [46]: indeed, there is a time delay  $\tau_{ne}$  during which the muscle is preparing to produce force, and once the muscle begins contracting, the tension ramps up to a peak that does not coincide with the EMG peak. One simple way to model this delay is using a first order differential equation [46] between the filtered EMG,  $e(t)$ , and the *muscle excitation* signal usually labeled  $u(t)$ , as follows:

$$\dot{e} = (u - e)/\tau_{ne} \quad (5)$$

where  $\tau_{ne}$  is the *excitation time constant*. Let us note that  $e(t)$  is still a number between 0 and 1, also representing the muscle fiber recruitment rate.

## Activation Dynamics

The muscle excitation  $u(t)$  can be related to the corresponding *muscle activation*  $a(t)$  by a non-linear first order differential equation [44]:

$$\dot{a} = (u - a)/\tau_a(a, u) \quad (6)$$

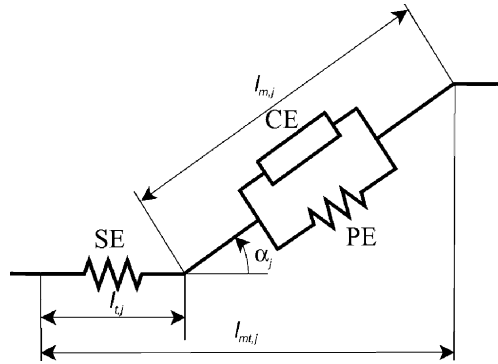
where  $\tau_a(a, u)$  is a time constant that varies with activation level and whether the muscle activation level is increasing or decreasing [38, 44]:

$$\tau_a(a, u) = \tau_{act}(0.5 + 1.5a) \quad \text{if } u \geq a \quad (7)$$

$$\tau_{deact}/(0.5 + 1.5a) \quad \text{if } u < a \quad (8)$$

where  $\tau_{act}$  is the *activation time constant* and  $\tau_{deact}$  is the *deactivation time constant*. Finally, the muscle activity is normalized with respect to the maximum voluntary

**Fig. 5** Representation of the Hill muscle model, including the contractile element (CE), the parallel elastic element (PEE) and the series elastic element (SEE).  $l_m$  denotes the muscle length,  $l_t$  the tendon length and  $l_{mt}$  the global musculo-tendon length



contraction activity [40, 44], so that  $a(t)$  is a number between 0 and 1, representing the muscle fiber recruitment rate.

### Contraction Dynamics

The transformation from raw EMG data to muscle force is partially based on the well-known *Hill model* [43, 44, 46] and is widely spread (e.g. [2, 8, 15, 23, 29, 38–40]), for which the musculo-tendon complex is composed of the tendon and the muscle (Fig. 5):

- The tendon – the elastic *Series Element* (SE) – is a “passive” wire that does not “generate” motion.
- The muscle consists in a parallel *Passive Element* (generally noted PE) with an active *Contractile Element* (generally noted CE) that generates contraction of the muscle controlled by neural excitation and then its lengthening or its shortening. Contractions of the muscle are assumed to be iso-volume [46], i.e. without volume change during contraction.

Further, the musculo-tendinous force  $F_{mt}$  is computed using the *contraction dynamics* equation:

$$F_{mt} = F_{max} \cdot \left( \underbrace{a(t) \tilde{F}_l^{CE}(\tilde{l}_m) \tilde{F}_v^{CE}(\tilde{v}_m)}_{active} + \underbrace{\tilde{F}_l^{PE}(\tilde{l}_m)}_{passive} + \underbrace{b_m \tilde{v}_m}_{damping} \right) \quad (9)$$

where

- $F_{max}$  is the maximal isometric force that a muscle can perform;  $F_{max}$  is evaluated here by tables [24].
- $a(t)$  is the *muscle activation* [44] obtained from (6), a number between 0 and 1 representing the muscle fiber recruitment rate.
- $\tilde{F}_l^{CE}(\tilde{l}_m)$ ,  $\tilde{F}_v^{CE}(\tilde{v}_m)$  and  $\tilde{F}_l^{PE}(\tilde{l}_m)$  respectively represent the *active force-length*, *active force-velocity* and *passive force-length* relations [38, 43, 46],

which are defined by the Hill model; let us note that the  $\sim$  sign above the variables means that they are normalized, i.e. the force components  $\widetilde{F}_l^{CE}$ ,  $\widetilde{F}_v^{CE}$ ,  $\widetilde{F}_l^{PE}$  are normalized with respect to  $F_{max}$ , the muscle length  $\widetilde{l}_m$  is normalized with respect to the muscle optimal fiber length  $L_m^{opt}$  (given by [24]) and the muscle velocity  $\widetilde{v}_m$  is normalized with respect to the muscle maximal contraction velocity  $v_{max}$  (given by [24]).

- $b_m$  is the damping factor, experimentally set to 0.1 by [25].

As the last step of the process,  $F_{mt}$  is computed using the *contraction dynamics* given by (9). In this equation, we can observe that  $a(t)$  is given by the activation dynamics, that  $\widetilde{l}_m$  and  $\widetilde{v}_m$  can be computed from the muscle kinematic measurements, and that  $b_m$  is set equal to 0.1 by [25]. Consequently, we will essentially describe in this section the formulations of the last elements of (9), i.e. the force components  $\widetilde{F}_l^{CE}$ ,  $\widetilde{F}_v^{CE}$  and  $\widetilde{F}_l^{PE}$ , on the basis of different references, as these formulations are continually improved:

1. The active force-length relationship  $\widetilde{F}_l^{CE}$  of the muscle is represented by a Gaussian function [38, 43]:

$$\widetilde{F}_l^{CE} = e^{-\left(\widetilde{l}_m - 1\right)^2 / \gamma} \quad (10)$$

where  $\widetilde{l}_m$  is the normalized muscle fiber length, and  $\gamma$  is a shape factor that approximates the force-length relationship of individual sarcomeres, set to 0.45 [38].

2. The active force-velocity relationship  $\widetilde{F}_v^{CE}$  of the muscle is represented by the following function [38, 43]:

$$\widetilde{F}_v^{CE} = -A_f \left( 1 + \frac{0.25 + 0.75a}{\widetilde{v}_m} \right) \quad \text{if } \widetilde{F}_v^{CE} \leq a\widetilde{F}_l^{CE} \quad (11)$$

$$\frac{\frac{\widetilde{v}_m(2 + 2/A_f)\widetilde{F}_m^{len}}{(0.25 + 0.75a)(\widetilde{F}_m^{len} - 1)} + 1}{\frac{\widetilde{v}_m(2 + 2/A_f)}{(0.25 + 0.75a)(\widetilde{F}_m^{len} - 1)} + 1} \quad \text{if } \widetilde{F}_v^{CE} > a\widetilde{F}_l^{CE} \quad (12)$$

where

- $a$  is the previously computed muscle activation.
- $\widetilde{v}_m$  is the normalized muscle contraction velocity.
- $\widetilde{F}_m^{len}$  is the maximum normalized muscle force achievable when the fiber is lengthening, set to 1.4 for young adults [38].
- $A_f$  is a force-velocity shape factor, which was set to 0.25 [44].

Let us note that (11) and (12) must be evaluated in a first time. Then the choice between the formulations of (11) and (12) can be made a posteriori by evaluating the inequalities between  $\widetilde{F}_v^{CE}$  and  $a\widetilde{F}_l^{CE}$ .

3. The passive force-length relationship  $\widetilde{F}_l^{PE}$  of the muscle is represented by an exponential function [38, 43]:

$$\widetilde{F}_l^{PE} = \frac{e^{k^{PE}(l^m - 1)/\varepsilon_m^0} - 1}{e^{k^{PE}} - 1} \quad (13)$$

where

- $k^{PE}$  is a shape factor, set to 5 [38].
- $\varepsilon_m^0$  is the passive muscle strain due to maximum isometric force, set to 0.6 for young adults [38].

Despite the fact that this transformation process is widely spread, it is not trivial because it is based on a model, on factors given by statistical tables, and on raw EMG data that do not directly measure the muscle activation but the external electromagnetic field generated by this activation.

## 2.4.2 Muscle Force Quantification

The method will be developed considering the consecutive steps introduced in Sect. 2.1.2, i.e. the kinematics identification that defines the model motion, the inverse dynamics that computes the elbow joint torque, and the proposed method to solve the muscle redundancy problem.

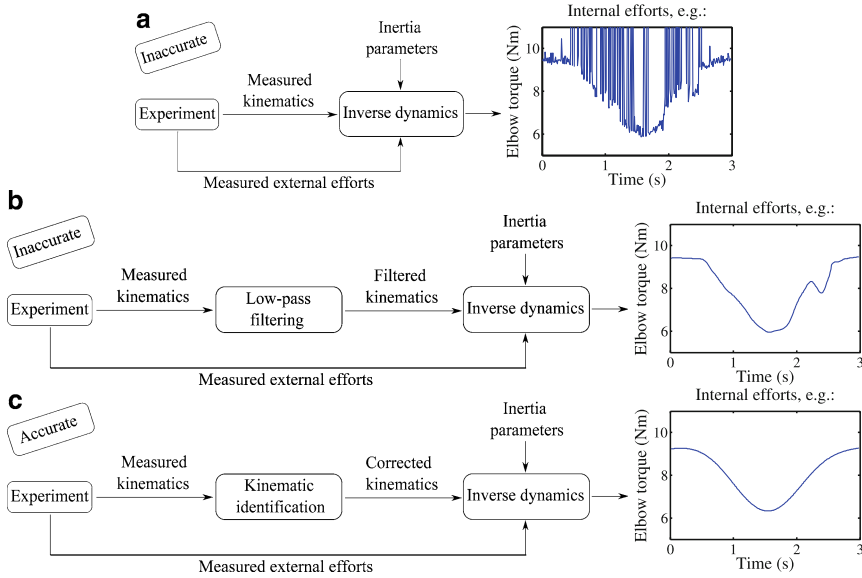
### Kinematics Identification

Let us note from the start that the inverse dynamics is a familiar tool to obtain results of joints efforts, but it is not obvious to obtain accurate results that could be usefully exploited, e.g. for the joint analysis of pathologic cases or the design of intelligent prostheses. This problem is illustrated in Fig. 6a during one cycle of forearm flexion/extension, for which the elbow joint torques have been computed using inverse dynamics, using raw measured kinematic and dynamometric data. Concerning this method, [9] and [10] indicate that the estimate of the internal forces is particularly sensitive to accelerations. Those are classically calculated by the method of finite differences starting from the measured positions, which dramatically amplifies the measurement errors.

Further, let us insist on the fact that filtering the undesired high frequencies in the measured kinematic data using a low-pass filter, and even smoothing the corresponding derivatives, i.e. velocities and accelerations, is not sufficient, as illustrated in Fig. 6b during the same cycle of forearm flexion/extension, for which the elbow joint torques have been computed using inverse dynamics.

Today, the most accurate way to obtain results of joint efforts is using inverse dynamics with corrected kinematics based on a kinematic identification process,





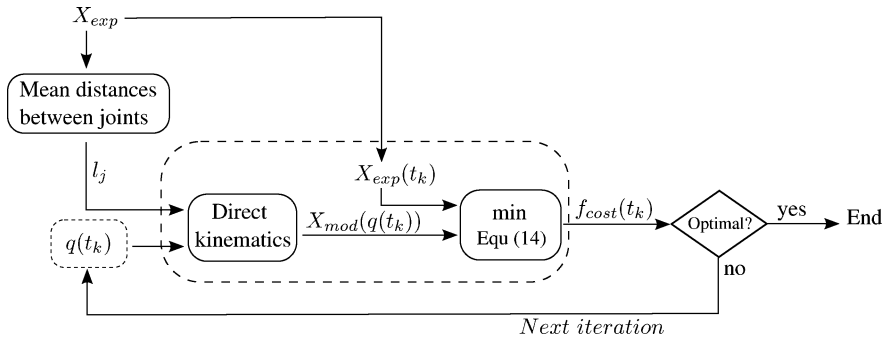
**Fig. 6** Schematic representation of inverse dynamics methods during one cycle of forearm flexion/extension: **(a)** Pure inverse dynamics, directly using the measured kinematics (totally inaccurate); **(b)** Inverse dynamics with filtered kinematics using a numerical low-pass filter (still inaccurate); **(c)** Inverse dynamics with corrected kinematics using a kinematic identification process (the most accurate way)

also known as solidification method [12], as illustrated in Fig. 6c. Concretely, this process estimates the joint coordinates of the MBS that best fit the experimental joint positions  $X_{exp,s}$ . This kinematic optimization can be formulated as a nonlinear least-square problem applied for each body configuration, at each time instant  $t_k$ ,  $k = 1, \dots, T$ , where  $T$  is the last time sample of each test. Consequently, the cost function  $f_{cost}(t_k)$  can be written at each time instant  $t_k$  as follows:

$$f_{cost}(t_k) = \sum_{s=1}^{n_{sens}} |X_{mod,s}(q(t_k)) - X_{exp,s}(t_k)|^2 \quad (14)$$

where:

- the index  $s = 1, \dots, n_{sens}$  indicates the optokinetic sensor. For instance for the upper limb,  $n_{sens} = 7$  as described in Fig. 3b).
- $q(t_k)$  is the vector of joint relative coordinates of the MBS at the time instant  $t_k$ , and is the variable of the optimization process.
- $X_{mod,s}(q(t_k))$  is the Cartesian coordinate of the  $s^{th}$  optokinetic sensor at time instant  $t_k$ , obtained from  $q(t_k)$ , using the direct kinematic model of the MBS.
- $X_{exp,s}(t_k)$  is the Cartesian coordinate of the  $s^{th}$  optokinetic sensor at time instant  $t_k$ , provided by the experimental set-up.



**Fig. 7** Optimization process for a body configuration, at a time instant  $t_k$

Figure 7 schematically outlines the optimization process, which involves two consecutive steps:

1. A pre-process calculates the mean distances  $l_j$  between the joints for each of the  $j$ th body segment, using the experimental joint Cartesian coordinates  $X_{exp,s}(t_k)$ . The reason is that the approach is based on an MBS composed of *rigid* bodies, for which a variable size of the bodies would be irrelevant.
2. The model joint Cartesian coordinates  $X_{mod,s}$  are given by a direct kinematic model using the  $l_j$  distances and an initial value (set to zero) of the joint coordinates  $q(t_k)$  that we want to determine. The cost function of this least-square optimization (14) is defined as the sum of the square components of the absolute error vector between  $X_{exp,s}(t_k)$  and  $X_{mod,s}(q(t_k))$  of the  $n_{sens}$  optokinetic sensors at the time instant  $t_k$ . In order to improve the numerical convergence, the optimal value of  $q(t_k)$  is obviously chosen as the initial condition of the next iteration at time instant  $t_{k+1}$ .

Finally, now that the configurations  $q$  are properly corrected, they can be filtered by precaution using a low-pass filter. Then, the corresponding velocities,  $\dot{q}$ , and accelerations,  $\ddot{q}$ , can be obtained via numerical differentiation techniques.

### Inverse Dynamics

The multibody dynamical equations are obtained from a Newton–Euler formalism [34]: this algorithm provides the vector  $Q_{inv}$  of internal interaction torques and forces at the joints for any configuration of the MBS, in the form of an inverse dynamical model (15), or a semi-direct dynamical model (16):

$$Q_{inv} = \phi(q, \dot{q}, \ddot{q}, F_{ext}, M_{ext}, g) \quad (15)$$

$$= M(q)\ddot{q} + c(q, \dot{q}, F_{ext}, M_{ext}, g) \quad (16)$$

where:

- $q$  ( $7 \times 1$ ) is the vector of the human body relative generalized coordinates, i.e. successively 3 arm rotations + 3 forearm rotations + 1 hand rotation<sup>1</sup> = 7 components.
- $\dot{q}$  and  $\ddot{q}$  ( $7 \times 1$ ) are the joint velocities and accelerations, respectively.
- $M(q)$  ( $7 \times 7$ ) is the generalized mass matrix.
- $c(q, \dot{q}, F_{ext}, M_{ext}, g)$  ( $7 \times 1$ ) is the dynamical vector containing the gyroscopic, centripetal, Coriolis terms as well as the external forces  $F_{ext}$  ( $7 \times 3$ ) and torques  $M_{ext}$  ( $7 \times 3$ ) and gravity  $g$  ( $1 \times 3$ ) applied to the system.

### Muscle Redundancy Solving

As introduced in Sect. 2.1.2, the key point of our muscle overactuation solving is based on an optimization process that computes the forces of the flexor and extensor muscle sets during forearm flexion/extension. Starting from the  $K$  and  $\Delta$  factors given by step 1 of the protocol and the filtered flexor/extensor forces during flexion/extension, this optimization adapts the  $K$  and  $\Delta$  factors of the flexors and extensors so that the corresponding elbow net torque  $Q_{emg}$  best fits the elbow net torque  $Q_{inv}$  given by the inverse dynamics, on the whole trajectory. Practically, this problem is formulated using the non-linear least squares method 'lsqnonlin' provide by Matlab.

## 3 Results

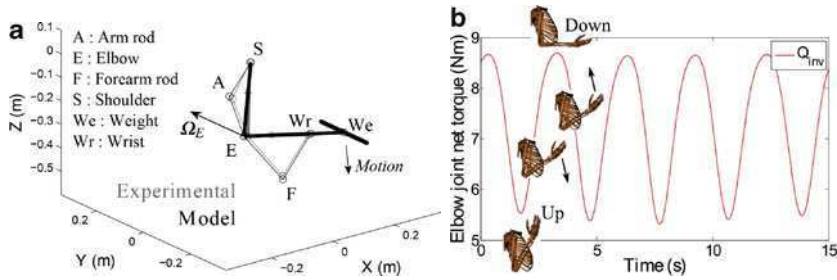
### 3.1 Joint Kinematics and Dynamics

Fig. 8a presents a comparison of the model configuration given by the experimental data and the resulting configuration obtained using the kinematics identification process. Figure 8a also features the corresponding direction of the instantaneous rotation velocity vector onto which the elbow joint torque is projected in order to obtain the elbow *net* joint torque  $Q_{emg}$ , corresponding to the torque component that contributes to the 'one DOF rotational' motion. Let us note that the other torque components also exist, but these ones contribute to the joint cohesion efforts (constraint torques).

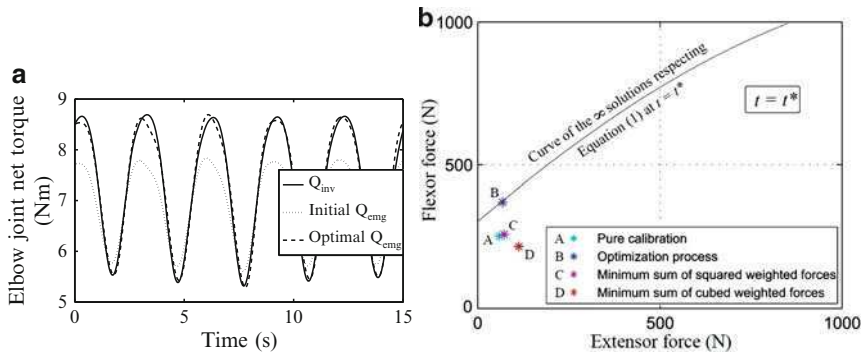
Figure 8b presents the time evolution of the corresponding computed elbow joint net torque  $Q_{inv}$  during the test, obtained via inverse dynamics.

---

<sup>1</sup> momentarily blocked.



**Fig. 8** Subject carrying a weight and performing several cycles of forearm flexion and extension: (a) comparison of the model configuration given by the experimental data and the resulting data from the kinematics identification process, and also featuring the weight and the corresponding direction of the instantaneous angular velocity vector  $\Omega_E$ ; (b) time evolution of the corresponding computed elbow joint net torque  $Q_{inv}$  during the test.



**Fig. 9** (a) Comparison between the elbow joint net torque  $Q_{inv}$  obtained from inverse dynamics, the equivalent joint net torque obtained from the calibrated Forces  $F_{f\&e,I}$  (output of protocol step 1) via (1), and the equivalent joint net torque obtained from the optimized Forces  $F_{f\&e,F}$  (output of protocol step 2) via (1); (b) Comparison of the solutions of the Biceps brachii and Triceps brachii set contributions at one time instant  $t = t^*$ : the solution of  $F_{f\&e,I}$  given by the pure calibration, the solution of  $F_{f\&e,F}$  given by our proposed optimization process, and the solutions of strategies that maximize endurance [13], i.e. that minimize the sum of the weighted forces at square or at cube

## 3.2 Muscle Forces

### 3.2.1 Muscle Force Assessment

First, Fig. 9a presents the time evolution of the elbow joint net torque  $Q_{inv}$  obtained from inverse dynamics, compared to the equivalent joint net torque obtained from the calibrated Forces  $F_{f\&e,I}$  (output of protocol step 1) via (1), and also compared to the equivalent joint net torque obtained from the optimized Forces  $F_{f\&e,F}$  (output of protocol step 2) via (1).

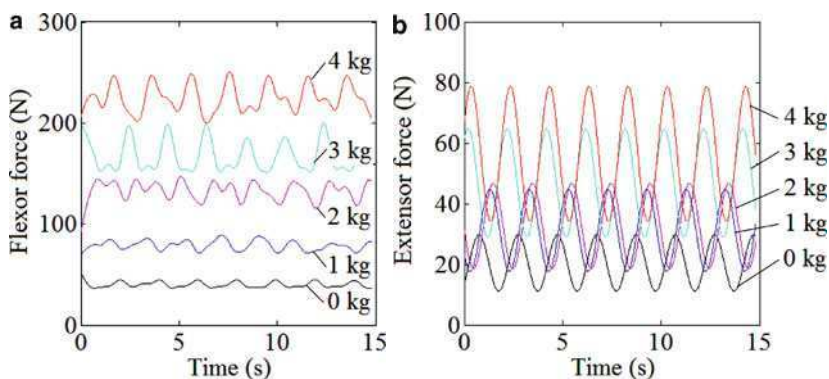
Secondly, Fig. 9b compares several solutions of the biceps brachii (flexors) and triceps brachii (extensors) set contributions at one time instant  $t$  of the flexion/extension:

1. The solution of  $F_{f\&e,I}$  given by the pure calibration (output of protocol step 1, and input of protocol step 2).
2. The solution of  $F_{f\&e,F}$  given by our proposed optimization process (output of protocol step 2), corresponding to the re-scaled and re-shifted calibrated value of  $F_{f\&e,I}$ .
3. The solution of a strategy that maximizes endurance [13], by minimizing the sum of the weighted forces at square or at cube.

### 3.2.2 Statistical Validation

For one subject, Fig. 10 presents the results of muscle forces during forearm flexion/extension, using the final values of the  $K$  and  $\Delta$  factors during the muscle overactuation solving process.

First of all, Figs. 3.2a and 3.2b clearly – and fortunately – show a gradation of the forces according to the weights carried. More fundamentally, a statistical validation of this muscle effort quantification method was performed with six male and six female subjects carrying five different weights (from 0 to 4 kg) with several flexion/extension frequencies ( $\frac{1}{2}$ ,  $\frac{1}{3}$  and  $\frac{1}{4}$  Hz). This validation shows a good inter-test reproducibility (also showing a fatigue effect) and a very good correlation (correlation factor  $r = 0.99$ ) between  $Q_{inv}$  and  $Q_{emg}$  at the end of the identification process.



**Fig. 10** Results of (a) the summed flexor (biceps brachii set) forces and (b) extensor (triceps brachii set) forces for one subject carrying weights from 0 to 4 kg and performing cycles of forearm flexion/extension at 0.5 Hz

## 4 Discussion

### 4.1 Joint Kinematics and Dynamics

The present inverse dynamical model of the human body (Fig. 1), necessarily preceded by a kinematic identification of the model configurations, is proposed as a satisfying method to estimate the joint efforts in dynamical context [31]. This problem being deterministic,  $Q_{inv}$  becomes a sufficiently accurate result that can be exploited as a *reference* for the optimization process that attempts to solve the muscle force redundancy (protocol step 2).

Nevertheless, let us point out that three main limitations of the present inverse dynamical model must be discussed in order to improve the process in the future:

1. *A geometrical limitation, due to the use of spherical joints:* The results of the kinematic analysis for this experiment show that the spherical joints considered here sufficiently fit the envisaged motion, with a mean absolute error on the Cartesian coordinates that is inferior to 3 mm in each direction at each joint (shoulder + elbow). However, using previous investigation results, in order to model more complex 3D motions, the present model will be extended to include more involved joints in the future, particularly to model shoulder [3] that is far from being a spherical joint.
2. *A kinematic limitation, due to the rigid MBS assumption:* Like other classical dynamical inverse analyses [17, 19, 36, 45] in Biomechanics of motion, the proposed model is composed of rigid segments. However, in reality, the body is not composed of a set of rigid bodies. Rather, each body member consists of a rigid part (bone), and a non-rigid part (skin, muscle, ligament, tendon, connective tissue, and other soft tissue structures) [26]: during any motion, the skeletal structure of the body experience accelerations, whereas the soft tissue motion is delayed, due to damped vibrations of the member. Consequently, the errors in the optimized joint coordinates  $q$  may introduce kinematical errors in the velocities  $\dot{q}$  and accelerations  $\ddot{q}$ , and thus introduce errors in the estimation of the internal efforts [31].
3. *A dynamical limitation, due to the approximation of the body inertia parameters:* The body inertia parameters, i.e. the masses, moments of inertia and centre of mass positions of the body members (the arm and the forearm) are approximated, using inertia tables [18]. Consequently, the errors in the estimated net joint efforts  $Q_{inv}$  increase if the corresponding body member accelerations increase. This is the reason why the present model is only proposed for rather low dynamics, such as our tests of forearm flexion/extension with a cycle frequency from  $\frac{1}{4}$  to  $\frac{1}{2}$  Hz, or other tests such as getting up from a seat [31], gait experiments [30] or other motions without significant dynamics or impact.

## 4.2 Muscle Force Quantification

If we make the assumption that the optimized kinematics and the net joint efforts  $Q_{inv}$  resulting from inverse dynamics are sufficiently accurate to attempt to solve the muscle overactuation problem (as discussed in Sect. 4.1), we still have to deal with two main problems [32] in order to estimate the muscle efforts:

1. the noisy-corrupted raw EMG signals; consequently, we will discuss in Sect. 4.2.1 the role of the EMG processing, including the Hill model and its numerous parameters, and the choice of parameters that we decided to vary in the optimization process (Sect. 2.4.2).
2. the undetermined number of solutions of muscular efforts; consequently in Sect. 4.2.2, we will compare our proposed solution with the main solutions of ‘maximum endurance’ strategy and ‘EMG calibration’ introduced in Sect. 1.2.

### 4.2.1 EMG Processing and Parameter Choice

The raw EMG signals are noisy-corrupted and do not provide accurate quantitative values of the actual muscle excitation  $u(t)$  and consequently of the actual muscle force  $F_m$ . However, we guess that this information, combined with a matured EMG processing (Sect. 2.4.1) and a muscle force calibration process can provide a first estimation of the muscle force that will constitute a reasonable initial value of the optimization process (Sect. 2.4.2), which will compute the predicted muscle forces during the motion.

Further, if we attempt to solve the muscle overactuation problem using an optimization process, “*too many parameters is not good*” (Buchanan et al. [8]): on the one hand, increasing the number of parameters to vary for a specific test shall certainly help to better fit the estimated joint torque  $Q_{emg}$  to the joint torque  $Q_{inv}$  obtained from inverse dynamics; but on the other hand, increasing the number of parameters may increase the sensitivity of the optimization process, and thus decrease the repeatability of the muscle force results from one test to another.

In consequence, we decided to use the EMG processing (Sect. 2.4.1), and more specifically the Hill model, as a black-box based on parameter tables, so that we will not have to deal with too many parameters in the optimization process, and we finally prefer to deal with the scaling  $K$  and the shifting  $\Delta$  factors of the muscle forces  $F_{f\&e,I}$  resulting from the Hill model.

### 4.2.2 Comparison to Existing Methods

When using an anatomical model of the musculoskeletal system, an infinite undetermined number of solutions of muscular efforts can correspond to the joint efforts  $Q_{inv}$ , because the joints are overactuated. In order to find the best physiologically

admissible solution, we decided to estimate the muscle forces using kinematic data, the  $Q_{inv}$  that becomes our “reference value”, and the information contained by the EMG signal.

The comparison between all considered solutions (Fig. 9b) shows that:

1. The solution of  $F_{f\&e,I}$  given by the pure calibration does not minimize  $Q_{inv} - Q_{emg}$  but, being not too far from the final solution, their role as initial values of our optimization process was useful.
2. The solution of  $F_{f\&e,F}$  given by our proposed optimization process minimizes  $Q_{inv} - Q_{emg}$  with a quantifiable difference between  $Q_{inv}$  and  $Q_{emg}$  (Fig. 9a) and this can be proposed as an admissible solution of muscle force repartition.
3. The solutions of strategies that maximize endurance [13] by minimizing the sum of the weighted forces at square or at cube do not seem to minimize  $Q_{inv} - Q_{emg}$ ; these are just approximation of the human motion strategies; further, we can notice that the solution of  $F_{f\&e,F}$  given by our proposed optimization is generally superior to the forces given by these strategies, which tends to confirm that the actual force repartition during motion does not always maximize endurance, and more generally varies from a strategy to another during the motion, as noticed Crowninshield et al. [13] who developed and discussed these strategy models.

To end, the statistical validation will show a good inter-test reproducibility (also showing a fatigue effect) and a very good correlation (correlation factor  $r = 0.99$ ) between  $Q_{inv}$  and  $Q_{emg}$  at the end of the identification process. This reinforces that our method of muscle force calibration plus optimization, based on muscle activity measurements, is a promising alternative to the methods based on the strategies usually used in the literature.

In summary, today, we can say that there are as many solutions of muscular efforts as methods that attempt to predict these. Nevertheless, the solution of  $F_{f\&e,F}$  given by our proposed protocol is coherent with the joint kinematics and system dynamics, while exploiting as well as possible the experimental information on the muscle activation.

### 4.3 Prospects

The main prospects of this research is to quantify with a satisfying accuracy the main muscle set efforts of subjects in different dynamical contexts, and to apply the model to develop:

- complementary diagnostic and follow-up tools for several pathologies such as lumbalgy, hemiplegy, scoliosis and pathologies that can hardly be diagnosed, such as fibromyalgy
- dimensioning tools for the design and choice of adapted prostheses by physicians and clinicians
- studies of effort hardness and ergonomic adaptation, e.g. when people with disabilities go into/out of vehicles



**Acknowledgements** The authors are grateful to Pr. P.Y. Willems, Emeritus of the Université catholique de Louvain, Louvain-la-Neuve, Belgium, for his help and support.

## References

1. Amarantini D (2003) Estimation des Efforts Musculaires À Partir des Données Périphériques: Application à l'Analyse de la Coordination Pluri-Articulaire. Thesis of the Université Joseph Fourier, Grenoble, France
2. Anderson FC, Pandy MG (2001) Dynamic optimization of human walking. *J Biomech Eng* 123(5):381–390
3. Bao H, Willems PY (1999) On the kinematic modelling and the parameter estimation of the human shoulder. *J Biomech* 32:943–950
4. Blajer W, Czaplicki A (2001) Modeling and inverse simulation of somersaults on the trampoline. *J Biomech* 34(12):1619–1629
5. Bouisset S (2002) Biomécanique et physiologie du mouvement. Abrégés, Editions Masson, Paris, France
6. Bouisset S, Le Bozec S, Ribreau C (2002) Postural dynamics in maximal isometric ramp efforts. *Biol Cybern* 87(3):211–219
7. BTS Bioengineering (2010). <http://www.btsbioengineering.com/>
8. Buchanan TS, Lloyd DG, Besier TF (2004) Neuromusculoskeletal modeling: estimation of muscle forces moments and movements measurements of neural command. *J Appl Biomech* 20:367–395
9. Cappozzo A, Leo T, Pedotti A (1975) A general computing method for the analysis of human locomotion. *J Biomech* 8:307–320
10. Challis JH, Kerwin DG (1996) Quantification of the uncertainties in resultant joint moments computed in a dynamic activity. *J Sports Sci* 14:219–231
11. Chenut X, Fiset P, Samin JC (2002) Recursive formalism with a minimal dynamic parametrization for the identification and simulation of multibody systems. Application to the Human Body. *Multibody Syst Dyn* 8:117–140
12. Chèze L, Fregly BJ, Dimnet J (1995) A solidification procedure to facilitate kinematic analyses based on video system data. *J Biomech* 28:879–884
13. Crowninshield RD, Brand RA (1981) A physiologically based criterion of muscle force prediction in locomotion. *J Biomech* 14(11):793–801
14. Debril JF, Pudlo P, El Menceur M, Gorce P, Lepoutre FX (2007) Human articulation efforts estimation in the automobile vehicle accessibility movement – A pilot study. In: Proceedings of the 1st international conference on digital human modeling, computer science, Beijing, China, 22–27 July
15. De Groote F, Pipeleers G, Demeulenaere B, Jonckers I, Spaepen P, Swevers J, De Schutter J (2006) A convex optimization approach to dynamic musculoskeletal analysis. In: Proceedings CD of the 6th national congress on theoretical and applied mechanics, Ghent, Belgium, 29–30 May
16. De Groote F, Pipeleers G, Jonckers I, Demeulenaere B, Swevers J, De Schutter J (2007) Physiology based inverse dynamic analysis of normal and hemiparetic gait. In: Proceedings of the 16th annual meeting of ESMAC, gait & posture 26(S17), Athens, Greece, 24–29 September
17. De Jalón G, Bayo E (1993) Kinematic and dynamic simulation of multibody systems: the real-time challenge. Springer, New York
18. De Leva P (1996) Adjustments to zatsiorsky-seluyanov's segment inertia parameters. *J Biomech* 29(9):1223–1230
19. Denoth J, Gruber K, Ruder H, Keppler M (1984) Forces and torques during sport activities with high accelerations. Perren SM, Schneider E (eds) Biomechanics current interdisciplinary research. Martinus Nijhoff Publishers, Dordrecht, Netherlands, pp 663–668

20. Dul J, Johnson GE, Shiavi R, Townsend MA (1984) Muscular synergism II. A minimum-fatigue criterion for load sharing between synergistic muscles. *J Biomech* 17(9):675–84
21. Epstein M, Herzog W (1998) Theoretical models of skeletal muscle. Wiley, Chichester, England
22. Gergersen CS, Hull ML (2003) Non-driving intersegmental knee moments in cycling computed using a model that includes three-dimensional kinematics of the shank/foot and the effect of simplifying assumptions. *J Biomech* 36:803–813
23. He J, Levine WS, Loeb GE (1991) Feedback gains for correcting small perturbations to standing posture. *IEEE T Automat Contr* 36:322–332
24. Holzbaur KRS, Murray WM, Delp SL (2005) A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control. *Ann Biomed Eng* 33(6):829–840
25. Lloyd DG, Bessier TF (2003) An emg-driven musculoskeletal model to estimate muscle forces and knee joint moment in vivo. *J Biomech* 36:765–776
26. Nigg BM, Herzog W (eds) (1999) *Biomechanics of the musculo-skeletal system*, 2nd edn. Chichester, England, and Wiley, New York
27. Penrod DD, Davy DT, Singh DP (1974) An optimization approach to tendon force analysis. *J Biomech* 7:123–129
28. Pérez M, Ausejo S, Pargada J, Suescun A, Celigüeta JT (2003) Application of multibody system analysis for the evaluation of the driver's discomfort. In: *Proceedings of the multibody dynamics*, Lisbon, Portugal, 1–4 July
29. Raasch CC, Zajac FE, Ma B, Levine WS (1997) Muscle coordination of maximum-speed pedaling. *J Biomech* 30(6):595–602
30. Raison M, Aubin CE, Detrembleur C, Fisette P, Samin JC (2008) Quantification of intervertebral efforts during walking: comparison between a healthy and a scoliotic subject. *Stud Health Tech Informat* 140:61–64
31. Raison M, Detrembleur C, Fisette P, Samin JC, Willems PY (2005) Determination of joint kinematics and dynamics in the human body: application to a subject getting up from a seat. In: *Proceedings of Ecomas thematic conference on advances in computational multibody dynamics*, Madrid, Spain, 21–24 June
32. Raison M, Detrembleur C, Fisette P, Samin JC, Willems PY (2006) Estimation of human muscular efforts using a model based optimization method. In: *Proceedings CD of the 7th national congress on theoretical and applied mechanics*, Mons, Belgium, 29–30 May
33. Raison M, Gaudes C, Le Bozec S, Willems PY (2007) Determination of joint efforts in the human body during maximum ramp pushing efforts. *J Biomech* 40(3):627–33
34. Samin JC, Fisette P (2003) *Symbolic modeling of multibody systems*. Kluwer, Dordrecht, Netherlands
35. Seireg A, Arvikar RJ (1973) A mathematical model for evaluation of force in lower extremities of the musculo-skeletal system. *J Biomech* 6:313–326
36. Silva M, Ambrósio J (2004) Sensitivity of the results produced by the inverse dynamic analysis of a human stride to perturbed input data. *Gait Posture* 19(1):35–49
37. Silva M, Ambrósio J, Pereira M (1997) A multibody approach to the vehicle and occupant integrated simulation. *Int J Crashworthines* 2(1):73–90
38. Thelen DG (2003) Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults. *Trans ASME* 125:70–77
39. Umberger BR, Gerritsen KGM, Martin PE (2003) A model of human energy expenditure. *Comput Meth Biomech Biomed Eng* 6(2):99–111
40. Venture G, Yamane K, Nakamura Y (2005) Identifying musculo-tendon parameters of human body based on the musculo-skeletal dynamics computation and Hill-Stroeve muscle model. In: *Proceedings of 5th IEEE-RAS international conference on humanoid robots*, Tsukuba, Japan, 5–7 December
41. Weber W, Weber E (1836) *Mechanik der Menschlichen Gehwerkzeuge*, Gottingen. Gottinger, Germany
42. Willems PY (ed) (1979) *Introduction à la mécanique*. Masson, Paris, France

43. Winters JM (1990) Hill-based muscle models: a systems engineering perspective. Winters JM, Woo SLY (eds) Multiple muscle systems: biomechanics and movement organization. Springer, New York
44. Winters JM (1995) An improved muscle-reflex actuator for use in large-scale neuromusculoskeletal models. *Ann Biomed Eng* 23:359–374
45. Zacher I (2004) Strength Based Discomfort Model of Posture and Movement. In: Proceedings of the SAE international digital human modelling conference, Rochester, Michigan, USA, 15–17 June
46. Zajac FE (1989) Muscle and tendon: properties, models, scaling and application to biomechanics and motor control. *Crit Rev Biomed Eng* 17(4):359–411

# Computing Time Reduction Possibilities in Multibody Dynamics

Thorsten Schindler, Markus Friedrich, and Heinz Ulbrich

**Abstract** This document discusses computing time reduction possibilities for the dynamic simulation of large nonsmooth multibody systems. Exemplary regarding the model of a pushbelt continuously variable transmission known from literature, simplifications are derived to examine the underlying problems. Usually one has to deal with the calculation of an adequate initial state and the reduction of the computational effort during integration. In three sections stationary belt models for the determination of an initial value are proposed, time step size enlargements due to implicit time integration schemes are analysed and the distribution of the main effort per time step on several central processing units by shared memory parallelisation is investigated.

## 1 Introduction

One area of research at the Institute of Applied Mechanics of the Technische Universität München is the field of nonsmooth multibody systems. These special mechanical systems basically include rigid bodies and in space discretised deformable bodies [25] in an hybrid way. They are additionally characterised by rigid unilateral and bilateral contacts as well as impacts, which lead to discrete jumps within the system's velocities. So the degree of freedom is not a constant function, but changes during the simulation process and determines a time-variant topology. Like described in [26] a unitary mathematical and numerical formulation based on measure differential equations (MDE) and projection constraint functions was processed in the last decades at different research institutes, summarised e.g. in [1,3,4],

---

T. Schindler (✉), M. Friedrich, and H. Ulbrich  
Institute of Applied Mechanics, Technische Universität München, Boltzmannstraße 15,  
85748 Garching, Germany  
e-mail: [thorsten.schindler@mytum.de](mailto:thorsten.schindler@mytum.de); [m.friedrich@mytum.de](mailto:m.friedrich@mytum.de); [ulbrich@amm.mw.tum.de](mailto:ulbrich@amm.mw.tum.de)

in [9, 17] and in [8, 14, 15]. It allows for the efficient integration even of industrial systems with large numbers of transitions [16] and avoids both high artificial stiffnesses and additional modelling errors due to regularised interactions.

The software for modelling and simulation of nonsmooth dynamical systems at the Institute of Applied Mechanics is called *MBSim* [10]. It is able to handle nearly arbitrary dynamical systems according to

$$\dot{q} = Y(q)u, \quad (1)$$

$$M(q)\dot{u} = h(q, u, t) + W(q)\lambda, \quad (2)$$

$$(q, u, \lambda, t) \in \mathcal{N} \quad (3)$$

with the transformation matrix  $Y$  for the generalised velocities  $u$ , the mass matrix  $M$  and the generalised contact directions  $W$  possibly depending on the positions  $q$ . The time  $t$  explicitly occurs in the right hand side  $h$  and in the algebraic constraints  $\mathcal{N}$  for the solution of the interaction reactions  $\lambda$ .

For special complex applications some improvements are necessary to enhance the behaviour of the underlying solution algorithms and to further reduce the effective computing time, significantly. One example of such challenging applications is the computational description of the pushbelt Continuously Variable Transmission (CVT) where an input and an output pulley as well as the pushbelt set up the variator of the transmission system (*left side* of Fig. 1). Thereby, each of the pulleys consists of a fixed and an axially moveable V-shaped sheave. The pushbelt is composed of approximately 400 elements which are guided by two ring packages of nine to twelve steel rings (*right side* of Fig. 1). The differential equation of a planar model has been derived and validated [11, 19]. Reference [20] presents the extension to spatial dynamics also indicating the numerical difficulties concerning the extremely increasing computing time in comparison to the planar case.

Within this framework, the current paper reproduces simplified three-dimensional CVT models to examine the underlying problems and deals in detail with computing time reduction possibilities in multibody dynamics.



**Fig. 1** Pushbelt variator and pushbelt with elements

## 2 Determination of a Suitable Initial Value

In the majority of industrially relevant cases for oscillatory mechanical problems the analysis of a stationary target state is of primary interest. So, it is preferable to calculate a suitable initial value for positions and velocities fulfilling a stationary equality of forces and torques. This at least reduces the decay time of high-frequency vibrations at the beginning of the time dependent simulation. The calculation of such an initial value depends on the specific simulation model. One possibility is to derive a further abstraction of the dynamic model equations. Within the thematic scope one can e.g. consider the various stationary continuous belt models [18, 21], which promise to save about 0.2 s unnecessary real simulation time. Though, implications for the computing time depend on the relationship of the complexity of the dynamic and the stationary model.

### 2.1 Kinematics

With Fig. 2, it is  $i_r = \frac{r_O}{r_I}$  the initial transmission ratio of the neutral fibre's output and input radius,  $t_L = \sqrt{\sin^2(\varphi) d_A^2 + d_{\text{align}}^2}$  the trum length as well as the arc lengths of the neutral fibre given by  $b_I = 2r_I(\pi - \varphi)$  and  $b_O = 2r_O\varphi$ . It holds

$$r_I - r_O = \cos(\varphi) d_A, \quad (4)$$

$$\tilde{l}_R = 2t_L + b_I + b_O \quad (5)$$

with axes distance  $d_A$  and belt length  $\tilde{l}_R$ . Thereby, the alignment  $d_{\text{align}}$  has to be calculated with the relative position of in- and output pulley again depending on the transmission ratio. The system of nonlinear equations (4)–(5) in the unknowns  $r_I$  and  $\varphi \in [0, 2\pi)$  is solved with a NEWTON-method using numerical JACOBIAN evaluations and the initial values

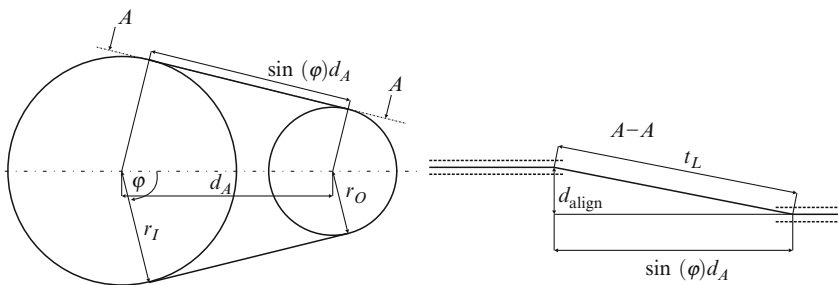


Fig. 2 Initialisation (axial view and view A-A)

$$r_{I_S} = (0.5 \tilde{l}_R - d_A) / \pi, \quad (6)$$

$$\varphi_S = \pi/2 \quad (7)$$

from a setting with  $i_r = 1$ . The results are used to define position values of the specific pushbelt CVT parts and to form a basis for the calculation of velocity estimates in the next sections.

## 2.2 Kinetics

In [21, Chapter 3.5] a planar velocity initialisation of a CVT is done successfully with a stationary kinetic belt model [12, 18]. In the following, the missing velocity description is derived in a similar way also not regarding outer plane effects. Concerning the belt model, the following assumptions have to be kept in mind:

- Planar continuum belt model with only longitudinal elasticity,
- Circle shaped pulley enlacement,
- Symmetrised wedge angle at the pulleys,
- Constant average COULOMB friction in the belt sheave contact,
- The influence of belt enlargement on radius change is neglected.

From the kinetic point of view the load torque induces different longitudinal forces in the trums because of friction forces between sheaves and belt. The difference of these forces compensates the excitation and is the same for the input and output pulley. First, the associated force propagation along an arc is derived; last, global estimations yield the final equations.

### 2.2.1 Geometry of the Excitation

Using Fig. 3 of an infinitesimal sheave sector with opening angle  $d\varphi$ , axis  $z$ , tangent  $t$  and radius direction  $r$  in the disk rotating coordinate system, it is  $\dot{g}$  the relative elastic slip velocity at the cone surface. The relative elastic slip velocity in the projection plane is  $\dot{g}_\star$  with  $\gamma$  being the angle of elastic slip,  $\delta$  the local half wedge angle including deformation and  $\delta_S$  its effective part. The decomposition of the friction force  $\mu dN$  in the axial, tangential and radial direction is given by

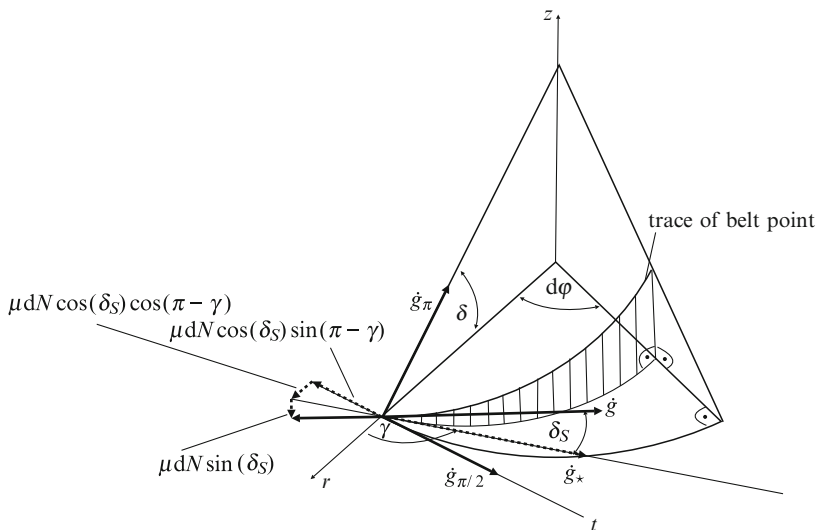
$$\mu dN \sin(\delta_S), \quad (8)$$

$$\mu dN \cos(\delta_S) \sin(\pi - \gamma), \quad (9)$$

$$\mu dN \cos(\delta_S) \cos(\pi - \gamma). \quad (10)$$

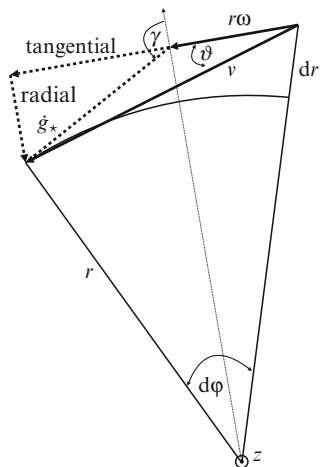
Comparing the direction of  $\dot{g}$  for  $\gamma = \pi$  given by  $\dot{g}_\pi$  and for  $\gamma = \pi/2$  given by  $\dot{g}_{\pi/2}$  yields

$$\tan(\delta_S) = -\tan(\delta) \cos(\gamma). \quad (11)$$



**Fig. 3** Belt slip at sheave

**Fig. 4** Absolute velocity



### 2.2.2 Relationship of Tangential and Radial Belt Velocity

In Fig. 4 one can see the trace of a belt point from axial point of view in a global coordinate system. So in addition to the relative velocities of Fig. 3, also the rotational velocity of the sheave  $r\omega$  is drawn as a positive initial velocity and the curvature angle  $\vartheta$  describing the change of the running radius  $r$  with respect to  $\varphi$  is defined. This results in formulas for the relative tangential and radial velocity



$$\dot{g}_{\star \tan} = v \cos(\vartheta) - r\omega \doteq v - r\omega, \quad (12)$$

$$\dot{g}_{\star \text{rad}} = \frac{dr}{dt} = r' \omega \quad (13)$$

with the linearised belt velocity

$$v \doteq \left[ 1 + \frac{L}{EA} \right] v_0 \quad (14)$$

depending on its undeformed part  $v_0$ , the longitudinal stiffness  $EA$  and the tensile force  $L$ . Together it is

$$\tan(\gamma) = -\frac{\dot{g}_{\star \tan}}{\dot{g}_{\star \text{rad}}} = -\frac{[EA + L]C_S - r}{r'} \quad (15)$$

with the elastic slip constant

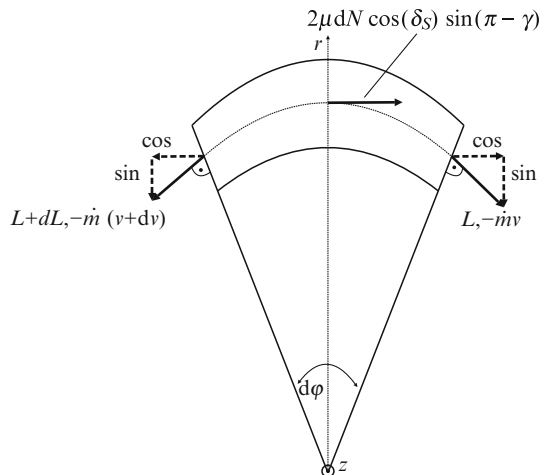
$$C_S := \frac{v_0}{EA\omega} \quad (16)$$

assuming  $\omega \neq 0$ .

### 2.2.3 Tangential Balance of Forces

Figure 5 shows the axial view to write down the equality of forces in tangential direction

$$[dL - \dot{m}dv] \cos(d\varphi/2) = 2\mu dN \cos(\delta_S) \sin(\pi - \gamma) \quad (17)$$



**Fig. 5** Axial view

involving the double tangential friction force from (9) on the right hand side because of a planar description and the constant stationary temporal mass distribution

$$\dot{m} = m^* v_0 \quad (18)$$

resulting from the local mass distribution  $m^*$  of the undeformed belt. This is needed both to simplify the equations and to improve their entropy according to [21, Chapter 3.5.1]. Because of (14), it is

$$dv = \frac{dL}{EA} v_0. \quad (19)$$

### 2.2.4 Radial Balance of Forces

Figure 6 can be used to write down the balance of forces in radial direction

$$[2L + dL - \dot{m} (2v + dv)] \sin(d\varphi/2) = 2dN [\sin(\delta) + \mu \cos(\delta_S) \cos(\pi - \gamma)]. \quad (20)$$

Thereby, the internal values appear on the left hand side, whereas the radial component of the friction from (10) and of the normal force contribute to the right hand side.

### 2.2.5 Axial Balance of Forces

Also the axial equilibrium of forces can be explained with Fig. 6. With the sheave expansion force  $S$  it is

$$dS = dN [\cos(\delta) - \mu \sin(\delta_S)]. \quad (21)$$

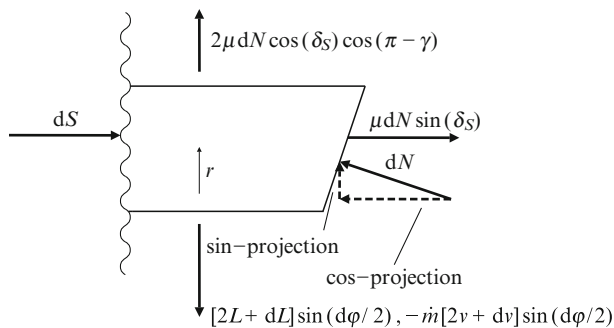


Fig. 6 Radial view

### 2.2.6 Change of the Running Radius

The equations (8)–(21) show the framework of [12, 18, 21] in its full generality. In contrast, changing of the running radius due to sheave tilting, transverse belt elasticity and axial sheave elasticity are not considered in the following because of their minor influence for the initialisation [21, Figs. 2.8 and 4.10]. These assumptions involve

$$\delta' \equiv 0, \quad r' \equiv 0, \quad \gamma \equiv \pm\pi/2, \quad \delta_S \equiv 0 \quad (22)$$

and avoid numerical difficulties mentioned in [21].

### 2.2.7 Summary of the Stationary Belt Model and Environment Interaction

From (17) and (20) it follows by the linearisations

$$\cos\left(\frac{d\varphi}{2}\right) \doteq 1, \quad \sin\left(\frac{d\varphi}{2}\right) \doteq \frac{d\varphi}{2}, \quad dL \frac{d\varphi}{2} \doteq 0, \quad (23)$$

some addition theorems and the consequences of the last paragraph

$$L' = \frac{dL}{d\varphi} = \frac{\pm\mu[L - \dot{m}v]}{[1 - \dot{m}\omega C_S] \sin(\delta_0)} = \pm \frac{\mu}{\sin(\delta_0)} L \mp \frac{\dot{m}\mu v_0 EA}{[EA - \dot{m}v_0] \sin(\delta_0)} \quad (24)$$

using the intermediate step

$$dL = \dot{m}dv + 2\mu dN \cos(\delta_S) \sin(\gamma) = \dot{m}dv \pm 2\mu dN, \quad (25)$$

$$Ld\varphi = \dot{m}vd\varphi + 2dN [\sin(\delta) - \mu \cos(\delta_S) \cos(\gamma)] = \dot{m}vd\varphi + 2dN \sin(\delta_0). \quad (26)$$

Equation (24) is an EULER-EYTELWEIN description of the longitudinal force propagation along an arc resulting in

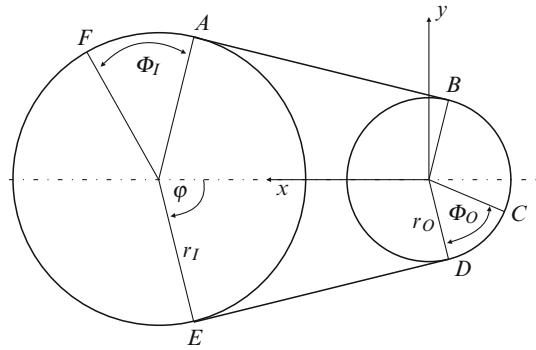
$$L(\varphi) = [L_0 - K] e^{\pm\mu^*(\varphi - \varphi_0)} + K \quad (27)$$

with

$$\mu^* := \frac{\mu}{\sin(\delta_0)}, \quad K := \frac{m^* v_0^2 EA}{EA - m^* v_0^2}. \quad (28)$$

If the denominator in the definition of  $K$  equals zero this would be a contradiction to (25) and (26).

The global belt setting is drawn in Fig. 7. The input arc  $E-A$ , the output arc  $B-D$ , the tight span  $D-E$ , the slack span  $A-B$ , the coordinate system and the

**Fig. 7** Belt kinematics

radii  $r_O$ ,  $r_I$  are canonical notations. The longitudinal force at the end of the input arc  $L_{I_{out}}$  occurs at point  $A$ . Similar,  $L_{I_{in}}$  can be found at  $E$ ,  $F$  and  $D$ ,  $L_{O_{in}}$  at  $B$  and  $C$  as well as  $L_{O_{out}}$  at  $D$ ,  $E$  and  $F$ . Generally, the longitudinal force increases at the output and decreases at the input arc only in an active part  $\Phi$ , but is always positive [21, Chapter 4.1.1]. When modelling elastic sheaves, the points  $F$  and  $C$  are called orthogonal points [18]. Altogether, this yields a consistent longitudinal force equivalence

$$L_{O_{in}} = L_{I_{out}} = (L_{O_{in}} - K) e^{\mu^*(\Phi_O - \Phi_I)} + K \quad (29)$$

resulting in

$$L_{O_{in}} = K \quad \vee \quad \Phi_O = \Phi_I. \quad (30)$$

Sticking implies two additional equations

$$r_I \omega_I = v_0 \left( 1 + \frac{L_{I_{in}}}{EA} \right) = v_0 \left[ 1 + \frac{(L_{O_{in}} - K) e^{\mu^* \Phi_O} + K}{EA} \right] \quad (\text{points } E - F), \quad (31)$$

$$r_O \omega_O = v_0 \left( 1 + \frac{L_{O_{in}}}{EA} \right) \quad (\text{points } B - C). \quad (32)$$

The equality of torques

$$M_O = r_O (L_{O_{out}} - L_{O_{in}}) = r_O (L_{O_{in}} - K) (e^{\mu^* \Phi_O} - 1) \quad (33)$$

with  $M_O$  being the positive load torque and the axial equality of forces

$$\begin{aligned}
 F_{CO} &= \int_{\varphi_O} S' d\varphi = \int_{\varphi_O} \frac{L [EA - m^* v_0^2] - m^* v_0^2 EA}{2 \tan(\delta_0) EA} d\varphi, \\
 &= \frac{EA - m^* v_0^2}{2 \tan(\delta_0) EA} \left[ \int_{-\varphi}^{\varphi - \Phi_O} L d\varphi + \int_{\varphi - \Phi_O}^{\varphi} L d\varphi \right] - \frac{m^* v_0^2}{\tan(\delta_0)} \varphi, \\
 &= \frac{EA - m^* v_0^2}{2 \tan(\delta_0) EA} \left[ 2L_{O_{in}} \varphi + (L_{O_{in}} - K) \left( \frac{e^{\mu^* \Phi_O} - 1}{\mu^*} - \Phi_O \right) \right] - \frac{m^* v_0^2}{\tan(\delta_0)} \varphi
 \end{aligned} \tag{34}$$

have to be solved concerning the output pulley. Hereby,  $F_{CO}$  is the positive axial clamping force at the output pulley as well as (26) and (21) yield the expression for  $S'$ .

### 2.2.8 Reduction of the Final Equations

Goal of this paragraph is the computation of  $\omega_O$ ,  $v_0$ ,  $\Phi_I$ ,  $\Phi_O$  and  $L_{O_{in}}$  for initialising velocity values. Of course, it is  $E > 0$ ,  $A > 0$ ,  $\delta_0 > 0$ ,  $\varphi > 0$ ,  $m^* > 0$ ,  $r_I > 0$ ,  $r_O > 0$ ,  $\omega_I > 0$  and so  $v_0 > 0$  for practical settings. Then clearly

$$L_{O_{in}} = \frac{EA r_O \omega_O}{v_0} - EA \tag{35}$$

according to (32). Concerning (30)–(34) there are two cases.

1. For  $L_{O_{in}} = K$  it follows necessarily  $M_O = 0$  and  $F_{CO} = 0$ . The condition  $F_{CO} = 0$  means that there is no sheave-belt contact, which is practically not relevant, and  $M_O = 0$  yields  $L_{O_{in}} = K$  or  $\Phi_O = 0$  from the mathematical viewpoint assuming  $\mu > 0$ . From the physical viewpoint obviously only  $\Phi_O = 0$  and so also  $\Phi_I = 0$  is interesting. So, it is

$$\omega_O = \frac{r_I}{r_O} \omega_I, \tag{36}$$

$$v_0 = \frac{\sqrt{(F_{CO} \tan(\delta_0) + EA \varphi)^2 + 4EA m^* r_I^2 \omega_I^2 \varphi^2} - EA \varphi - F_{CO} \tan(\delta_0)}{2m^* r_I \omega_I \varphi}. \tag{37}$$

2. If  $M_O \neq 0$  it is necessarily  $\Phi_I = \Phi_O$  and  $L_{O_{in}} \neq K$ . So,

$$\Phi_O = \frac{1}{\mu^*} \ln \left[ \frac{M_O}{r_O (L_{O_{in}} - K)} + 1 \right] \tag{38}$$

being only defined for  $M_O > r_O (K - L_{O_{in}})$ . This condition depends on the kinematic and kinetic setting and states the physical application in a same way as the inequalities

$$\Phi_I < 2(\pi - \varphi), \quad \Phi_O < 2\varphi \quad (39)$$

define the active arcs. Equations (31) and (34) remain only depending on  $\omega_0$  and  $v_0$ . They can be solved by a generalised NEWTON method with numerical JACOBIAN evaluation and starting values from the  $M_O = 0$  case.

Now,  $\omega_I$  and  $\omega_O$  can be used for the initialisation of the pulleys' and ring packages' angular velocity. According to (14) and (27) the expression

$$v = \left[ 1 + \frac{\left( EA \left( \frac{v_{in}}{v_0} - 1 \right) - K \right) e^{\mu^*(\varphi - \varphi_0)} + K}{EA} \right] v_0 \quad (40)$$

explains the behaviour of the ring package belt velocity in an active arc starting from  $v_{in}$  at  $\varphi = \varphi_0$ . Element velocities have to be inherited from the ring packages.

### 3 Computational Effort During Integration

The computational amount during the integration of a differential equation can be divided in the amount per time step and the number of time steps as a multiplier. In the following the spatial simulation of a flexible belt with rigid elements linearly arranged is discussed as a benchmark problem (cp. Fig. 8). The main effort per time step is defined by the kinematic element and finite element update and the contact behaviour; the time step size itself is declared by the numerical stiffness of the flexible part. As the contact iterations cannot be simplified conceptually, in the following the element update loop parallelisation and the ring package model stabilisation are analysed to reduce the overall CPU time.



**Fig. 8** Rigid elements with internal unilateral contacts on flexible belt

### 3.1 Stabilising Equations of Motion

According to [20] the flexible part is divided in  $\tilde{N}_B$  spatial large deformation beams giving a lot of modelling possibilities available in the literature [24]. As a co-rotational [6] model in [25] shows an efficient behaviour in the planar case compared to ANC formulations [7, 13, 22] it is extended to three dimensions using inertial approaches [2, 23] and applying the physically interpretable EULER-BERNOULLI beam formulation. The mathematical derivation is based on the ideas of finite element theory for assembling and multibody formulations for the evaluation of the equations of motion for each finite element.

#### 3.1.1 Coordinate Settings

Using a general stationary frame of reference, the entire kinematic of one finite element can be described like in Fig. 9. Using a reversed Cardan parameterisation

$$\varphi_0(x) := \varphi_{S_0} + w'_0(x), \quad \varphi_1(x) := \varphi_{S_1} + w'_1(x), \quad \varphi_2(x) := \varphi_{S_2} + w'_2(x) \quad (41)$$

a set of *internal coordinates*

$$\mathbf{q}_i := (x_S, y_S, z_S, \varphi_{S_0}, \varphi_{S_1}, \varphi_{S_2}, \tilde{\varepsilon}, d_{L1}, d_{R1}, \beta_{L1}, \beta_{R1}, d_{L2}, d_{R2}, \beta_{L2}, \beta_{R2}, \kappa_0)^T \quad (42)$$

is defined by the position vector and the angle parameterisation of the trihedron of the finite element centre as well as the longitudinal strain, the coefficients

$$w_i(-l_0/2) := d_{Li}, \quad w_i(0) := 0, \quad w_i(l_0/2) := d_{Ri}, \quad i = 1, 2, \quad (43)$$

$$w'_i(-l_0/2) := \beta_{Li}, \quad w'_i(0) := 0, \quad w'_i(l_0/2) := \beta_{Ri}, \quad i = 1, 2 \quad (44)$$

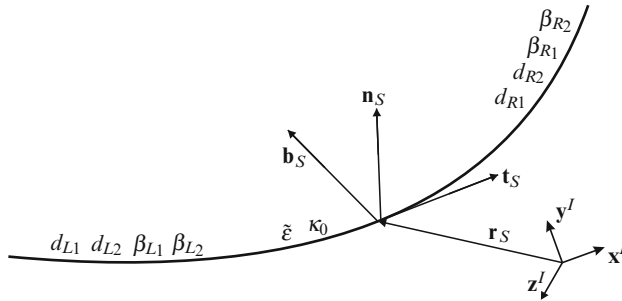


Fig. 9 Internal coordinates

of the ansatz functions with the finite element length  $l_0$  and the torsion

$$\kappa_0 := {}_I \mathbf{b} \cdot {}_I \mathbf{n}' = w_0'' - \sin(\varphi_{S_1}) w_2'' . \quad (45)$$

The degree of the polynomials

$$w_i := a_{w_i} x^5 + b_{w_i} x^4 + c_{w_i} x^3 + d_{w_i} x^2, \quad i = 0, 1, 2 \quad (46)$$

is a compromise between too much stiffening for lower order and too much support for higher order with the coefficients of  $w_0$  being constrained by the constant torsion characteristics of (45). Altogether, rigid and elastic body motion are decoupled and a compact form of the equations of motion with appropriate approximation not depending on the boundary conditions is available for evaluation.

For coupling of finite elements the global coordinates

$$\mathbf{q}_g := (x_L, y_L, z_L, \varphi_{L_0}, \varphi_{L_1}, \varphi_{L_2}, c_{L_1}, c_{R_1}, c_{L_2}, c_{R_2}, x_R, y_R, z_R, \varphi_{R_0}, \varphi_{R_1}, \varphi_{R_2})^T \quad (47)$$

with

$$c_{L_1} := w_1(-l_0/4), \quad c_{R_1} := w_1(l_0/4), \quad c_{L_2} := w_2(-l_0/4), \quad c_{R_2} := w_2(l_0/4) \quad (48)$$

are used (cp. Fig. 10) to get equations of motion in minimal representation. The information between the coordinate sets is transferred by the motion of the neutral fibre

$${}_I \mathbf{r}(x) = (1 + \tilde{\epsilon}) \int {}_I \mathbf{t} dx \doteq {}_I \mathbf{r}_S + (1 + \tilde{\epsilon}) x {}_I \mathbf{t}_S + \hat{w}_1(x) {}_I \mathbf{n}_S + \hat{w}_2(x) {}_I \mathbf{b}_S \quad (49)$$

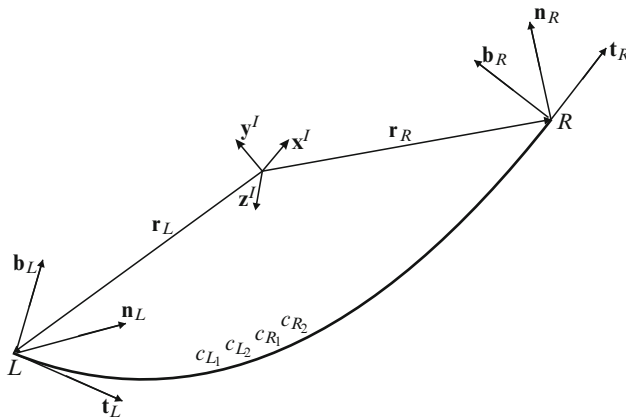


Fig. 10 Global coordinates



parametrised by the Lagrangian coordinate  $x$  with

$$\hat{w}_1 := \xi_{\tilde{n}} w_1 + \xi_{\tilde{b}} w_2, \quad \hat{w}_2 := \eta_{\tilde{n}} w_1 + \eta_{\tilde{b}} w_2. \quad (50)$$

This results in a transformation  $\mathbf{F}(\mathbf{q}_i, \mathbf{q}_g) = \mathbf{0}$ . One part can be solved analytically with respect to the internal coordinates:

$$\begin{aligned} {}_I \mathbf{r}_S = & \frac{{}_I \mathbf{r}_L + {}_I \mathbf{r}_R}{2} - \frac{[\xi_{\tilde{n}} (d_{L_1} + d_{R_1}) + \xi_{\tilde{b}} (d_{L_2} + d_{R_2})] {}_I \mathbf{n}_S}{2} \\ & - \frac{[\eta_{\tilde{n}} (d_{L_1} + d_{R_1}) + \eta_{\tilde{b}} (d_{L_2} + d_{R_2})] {}_I \mathbf{b}_S}{2} \end{aligned} \quad (51)$$

and

$$\tilde{\varepsilon} = \frac{1}{l_0} ({}_I \mathbf{r}_R - {}_I \mathbf{r}_L) \cdot {}_I \mathbf{t}_S - 1, \quad (52)$$

$$\kappa_0 = \frac{1}{l_0} [\varphi_{R_0} - \varphi_{L_0} - \sin(\varphi_{S_1}) (\beta_{R_2} - \beta_{L_2})]. \quad (53)$$

A system of nonlinear equations

$$\tilde{\mathbf{F}}_1 := \varphi_{S_0} - \frac{\varphi_{L_0} + \varphi_{R_0}}{2} + \sin(\varphi_{S_1}) \frac{\beta_{L_2} + \beta_{R_2}}{2} = 0, \quad (54)$$

$$\tilde{\mathbf{F}}_2 := \varphi_{S_1} - \frac{\varphi_{L_1} + \varphi_{R_1}}{2} + \frac{\beta_{L_1} + \beta_{R_1}}{2} = 0, \quad (55)$$

$$\tilde{\mathbf{F}}_3 := \varphi_{S_2} - \frac{\varphi_{L_2} + \varphi_{R_2}}{2} + \frac{\beta_{L_2} + \beta_{R_2}}{2} = 0, \quad (56)$$

$$\tilde{\mathbf{F}}_4 := \beta_{R_1} - \beta_{L_1} - \varphi_{R_1} + \varphi_{L_1} = 0, \quad (57)$$

$$\tilde{\mathbf{F}}_5 := \beta_{R_2} - \beta_{L_2} - \varphi_{R_2} + \varphi_{L_2} = 0, \quad (58)$$

$$\tilde{\mathbf{F}}_6 := \xi_{\tilde{n}} (d_{R_1} - d_{L_1}) + \xi_{\tilde{b}} (d_{R_2} - d_{L_2}) - (\mathbf{r}_R - {}_I \mathbf{r}_L) \cdot {}_I \mathbf{n}_S = 0, \quad (59)$$

$$\tilde{\mathbf{F}}_7 := \eta_{\tilde{n}} (d_{R_1} - d_{L_1}) + \eta_{\tilde{b}} (d_{R_2} - d_{L_2}) - ({}_I \mathbf{r}_R - {}_I \mathbf{r}_L) \cdot {}_I \mathbf{b}_S = 0, \quad (60)$$

$$\tilde{\mathbf{F}}_8 := 2b_{w_1} l_0^4 / 256 + 2d_{w_1} l_0^2 / 16 - c_{R_1} - c_{L_1} = 0, \quad (61)$$

$$\tilde{\mathbf{F}}_9 := 2a_{w_1} l_0^5 / 1024 + 2c_{w_1} l_0^3 / 64 - c_{R_1} + c_{L_1} = 0, \quad (62)$$

$$\tilde{\mathbf{F}}_{10} := 2b_{w_2} l_0^4 / 256 + 2d_{w_2} l_0^2 / 16 - c_{R_2} - c_{L_2} = 0, \quad (63)$$

$$\tilde{\mathbf{F}}_{11} := 2a_{w_2} l_0^5 / 1024 + 2c_{w_2} l_0^3 / 64 - c_{R_2} + c_{L_2} = 0 \quad (64)$$

remains which can be solved with NEWTON's method using analytical JACOBIAN evaluations. The derivatives fulfill the relations

$$\dot{\mathbf{q}}_i = \frac{d\mathbf{q}_i}{d\mathbf{q}_g} \dot{\mathbf{q}}_g =: \mathbf{J}_{ig} \dot{\mathbf{q}}_g, \quad (65)$$

$$\ddot{\mathbf{q}}_i = \frac{d}{dt} \left( \frac{d\mathbf{q}_i}{d\mathbf{q}_g} \right) \dot{\mathbf{q}}_g + \frac{d\mathbf{q}_i}{d\mathbf{q}_g} \ddot{\mathbf{q}}_g =: \dot{\mathbf{J}}_{ig} \dot{\mathbf{q}}_g + \mathbf{J}_{ig} \ddot{\mathbf{q}}_g \quad (66)$$

with the expressions involving the internal coordinates  $\mathbf{x}_{be}$  of the implicit relationship being calculated by the chain rule

$$\frac{\partial \tilde{\mathbf{F}}}{\partial \mathbf{x}_{be}} \frac{d\mathbf{x}_{be}}{d\mathbf{q}_g} = - \frac{\partial \tilde{\mathbf{F}}}{\partial \mathbf{q}_g}, \quad (67)$$

$$\frac{\partial \tilde{\mathbf{F}}}{\partial \mathbf{x}_{be}} \frac{d}{dt} \left( \frac{d\mathbf{x}_{be}}{d\mathbf{q}_g} \right) = - \frac{d}{dt} \left( \frac{\partial \tilde{\mathbf{F}}}{\partial \mathbf{q}_g} \right) - \frac{d}{dt} \left( \frac{\partial \tilde{\mathbf{F}}}{\partial \mathbf{x}_{be}} \right) \frac{d\mathbf{x}_{be}}{d\mathbf{q}_g}. \quad (68)$$

### 3.1.2 Equations of Motion

The bending length

$$l_b := \int_{-l_0/2}^{l_0/2} \| {}_I \mathbf{r}' \| dx \approx (1 + \tilde{\varepsilon}) l_0 + \frac{1}{2} \left[ \int_{-l_0/2}^{l_0/2} \hat{w}'_1 \hat{w}'_1 dx + \int_{-l_0/2}^{l_0/2} \hat{w}'_2 \hat{w}'_2 dx \right] \quad (69)$$

contains second order terms concerning bending and so allows for geometric non-linear foreshortening. The corresponding strain is given by

$$\varepsilon := \frac{l_b - l_0}{l_0} \approx \tilde{\varepsilon} + \frac{1}{2l_0} \left[ \int_{-l_0/2}^{l_0/2} \hat{w}'_1 \hat{w}'_1 dx + \int_{-l_0/2}^{l_0/2} \hat{w}'_2 \hat{w}'_2 dx \right]. \quad (70)$$

Considering mass conservation and at most quadratic elastic deformation terms yields for the gravitational, elastic and kinetic energy

$$\begin{aligned} V_g &= -\tilde{\rho}_R \tilde{A}_R {}_I \mathbf{g} \cdot \int_{-l_0/2}^{l_0/2} {}_I \mathbf{r} dx \\ &= -\tilde{\rho}_R \tilde{A}_R {}_I \mathbf{g} \cdot \left[ l_0 {}_I \mathbf{r}_S + \int_{-l_0/2}^{l_0/2} \hat{w}_1 dx {}_I \mathbf{n}_S + \int_{-l_0/2}^{l_0/2} \hat{w}_2 dx {}_I \mathbf{b}_S \right], \end{aligned} \quad (71)$$

$$\begin{aligned} V_e &\approx \frac{\tilde{E}_R \tilde{A}_R}{2} \varepsilon^2 l_0 + \frac{\tilde{E}_R \tilde{I}_1}{2} \int_{-l_0/2}^{l_0/2} (\hat{w}_1'' - \tilde{\kappa}_{10})^2 dx \\ &\quad + \frac{\tilde{E}_R \tilde{I}_2}{2} \int_{-l_0/2}^{l_0/2} (\hat{w}_2'' - \tilde{\kappa}_{20})^2 dx + \frac{\tilde{G}_R \tilde{I}_0}{2} \int_{-l_0/2}^{l_0/2} \kappa_0^2 dx, \end{aligned} \quad (72)$$

$$T \approx \frac{1}{2} \tilde{\rho}_R \left[ \tilde{A}_R \int_{-l_0/2}^{l_0/2} \|{}_I \dot{\mathbf{r}}\|^2 dx + \tilde{I}_0 \int_{-l_0/2}^{l_0/2} \omega_t^2 dx \right] \quad (73)$$

with initial curvatures  $\tilde{\kappa}_{1_0}$  and  $\tilde{\kappa}_{2_0}$  as well as the projection of the angular velocity on the local tangent  $\omega_t = \dot{\varphi}_0 - \sin(\varphi_1) \dot{\varphi}_2$  neglecting angular bending dependencies in the kinetic energy terms. The standard parameters are the density  $\tilde{\rho}_R$ , the cross-section  $\tilde{A}_R$ , Young's modulus  $\tilde{E}_R$ , shear modulus  $\tilde{G}_R$ , the area moments of inertia  $\tilde{I}_1$ ,  $\tilde{I}_2$ ,  $\tilde{I}_0$  and the gravity  ${}_I \mathbf{g}$ .

With the LAGRANGE II formalism

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\mathbf{q}}_i} \right)^T - \left( \frac{\partial T}{\partial \mathbf{q}_i} \right)^T + \left( \frac{\partial (V_e + V_g)}{\partial \mathbf{q}_i} \right)^T = \mathbf{0} \quad (74)$$

it is possible, to derive the equations of motion. Because of  $T = T(\mathbf{q}_i, \dot{\mathbf{q}}_i)$  it holds

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\mathbf{q}}_i} \right)^T = \frac{\partial^2 T}{\partial \dot{\mathbf{q}}_i^2} \ddot{\mathbf{q}}_i + \frac{\partial^2 T}{\partial \dot{\mathbf{q}}_i \partial \mathbf{q}_i} \dot{\mathbf{q}}_i. \quad (75)$$

Hence, the mass matrix and the smooth right hand side are given by

$$\mathbf{M}_i := \frac{\partial^2 T}{\partial \dot{\mathbf{q}}_i^2}, \quad \mathbf{h}_i := \left( \frac{\partial T}{\partial \mathbf{q}_i} \right)^T - \left( \frac{\partial (V_e + V_g)}{\partial \mathbf{q}_i} \right)^T - \frac{\partial^2 T}{\partial \dot{\mathbf{q}}_i \partial \mathbf{q}_i} \dot{\mathbf{q}}_i \quad (76)$$

such that

$$\mathbf{M}_i \ddot{\mathbf{q}}_i - \mathbf{h}_i = \mathbf{0}. \quad (77)$$

Globally the equations of motion fulfill

$$\underbrace{\mathbf{J}_{ig}^T \mathbf{M}_i \mathbf{J}_{ig}}_{\mathbf{M}_g} \ddot{\mathbf{q}}_g - \underbrace{\mathbf{J}_{ig}^T (\mathbf{h}_i - \mathbf{M}_i \mathbf{J}_{ig} \dot{\mathbf{q}}_g)}_{\mathbf{h}_g} = \mathbf{0}. \quad (78)$$

### 3.1.3 Analysis of Instability

When increasing the number of finite elements in an explicit integration scheme it has been recognized that NEWTON's method does not succeed in the solution of the transformation between internal and global coordinates although an analytical JACOBIAN and double machine precision are used for the nonlinear equations' solver (cf. Sect. 3.1.1). A solution can be found by decreasing the integrator time step size or using a linear implicit scheme with the necessary evaluation of finite element JACOBIAN matrices of the right hand side being the drawback. To avoid this additional effort the problem is analysed by reduction using norm consistent

linearisation of the transformation in the  $\mathbf{x}^I - \mathbf{y}^I$ -plane. This changes the following equations from Sect. 3.1.1

$$\tilde{\mathbf{F}}_1 := \varphi_{S_0} - \frac{\varphi_{L_0} + \varphi_{R_0}}{2} = 0, \quad (79)$$

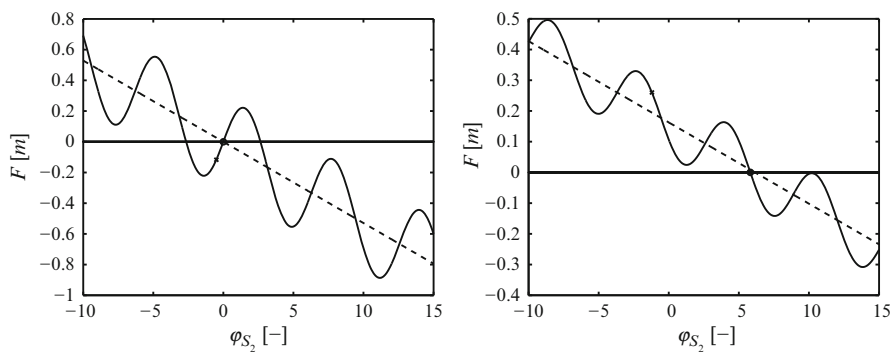
$$\tilde{\mathbf{F}}_6 := d_{R_2} - d_{L_2} + (x_R - x_L) \sin(\varphi_{S_2}) - (y_R - y_L) \cos(\varphi_{S_2}) = 0, \quad (80)$$

$$\tilde{\mathbf{F}}_7 := d_{L_1} - d_{R_1} - (z_R - z_L) = 0 \quad (81)$$

and yields a decoupling of the spatial motion. Finally, it results a nonlinear equation

$$F(\varphi_{S_2}) := (x_R - x_L) \sin(\varphi_{S_2}) - (y_R - y_L) \cos(\varphi_{S_2}) + \frac{64}{17} \left[ \frac{3l_0}{64} \left( \frac{\varphi_{L_2} + \varphi_{R_2}}{2} - \varphi_{S_2} \right) - c_{L_2} + c_{R_2} \right] = 0 \quad (82)$$

in  $\varphi_{S_2}$  comprising an affine equation and a superposed oscillation. Figure 11 shows this nonlinear function and its affine part in different settings. The solution of the last iteration and so the canonical starting value (*marked with a cross*) is always in the antinode around zero. If the integrator time step size is small enough (*left figure*) this yields a new solution (*marked with a circle*). Otherwise (*right figure*) e.g. when global input parameters (the axis intercept) blow up for increasing simulation time, the starting value for NEWTON's method would have to be shifted to the antinode of the expected solution. This instability depends on the oscillation projected to the abscissa and is typical with explicit integration schemes. The time step size for the explicit integration scheme  $\Delta t < 5 \cdot 10^{-8}$  s is defined by the numerical stiffness of the ring package equations of motion and not by  $\Delta t < 5 \cdot 10^{-6}$  s resulting from the Nyquist-Shannon sampling theorem and 20,000 Hz being the polygonal frequency upper bound of a pushbelt CVT. The theoretically possible improvement factor 100



**Fig. 11** Zeros of the planar beam transformation root function

in practice reduces to a maximum realistic improvement factor depending on the computer architecture because of the additional effort per time step of linear implicit integration schemes.

### 3.2 Parallel Computing Architectures

Today, Moore's law states that the number of transistors on a standard processor doubles every 18 months. Based on empirical studies, Gordon Moore formulated this rule of thumb in 1965 and proposed a validity period of about ten to 15 years in 2007. This yields a profit in performance, which cannot any more achieved by higher clock rates due to technical limitations but by parallelism on CPU level with multi-core architectures. The consequences for software developers are the adaptation of existing programs and the design of new ones concerning these hardware trends.

In the case of multibody systems normally one has no memory limits, such that a multiple instruction multiple data (MIMD) architecture with shared memory can be used. This is the field of the OPENMP interface [5] giving the possibility to simply extend a serial program with control structures for parallelisation. Then, the advantages can be measured by

$$\text{speed-up } s_n := \frac{t_1}{t_n}, \quad (83)$$

$$\text{efficiency } e_n := \frac{s_n}{n} \quad (84)$$

whereby  $t_n$  is the run-time of the program on  $n$  processors. There exist several theoretical estimates for the maximum achievable speed-up,

$$\text{Amdahl's law } s_n \leq \frac{1}{\sigma + \frac{1-\sigma}{n}} \leq \frac{1}{\sigma}, \quad (85)$$

$$\text{Gustafson's law } s_n = \frac{t_s + nt_p}{t_s + t_p} \leq n. \quad (86)$$

Both rules are based on different ideas. Amdahl assumes the serial code part  $\sigma$  to be constant when considering one simulation model. This results in an upper bound for the speed-up because of administrative overhead when enlarging the number of processors. Gustafson looks at the simulation time of a parallel program on a single core machine with the sequential simulation time part  $t_s$  and the parallel simulation time part  $t_p$ . For an asymptotic consideration he assumes the sequential simulation time to relatively decrease if one uses more processors e.g. in the practically relevant applications of larger simulation models. Altogether, one has a more pessimistic rule of Amdahl and a more optimistic one of Gustafson available for comparison with experimental scaling measurements of a concrete program.

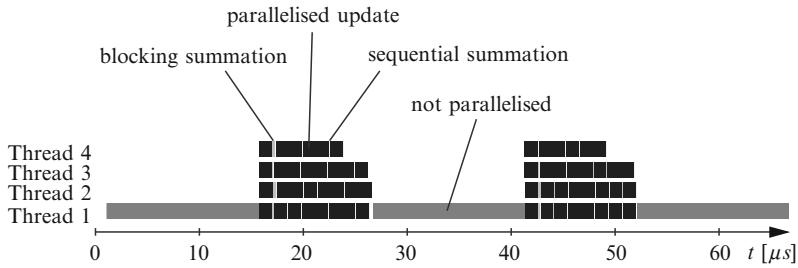


Fig. 12 Program flow

Profiling yields that in the discussed multibody system context of Fig. 8 the main computational cost per time step is spent for the kinematic update of the rigid elements and of the finite elements, potentially being used for the discretisation of the flexible belt. Both items are organised in loops from the software development point of view. This can be summarised by the formulas for the components of the global equations of motion

$$\mathbf{M} = \sum_i \mathbf{J}_i^T \mathbf{M}_i \mathbf{J}_i, \quad \mathbf{h} = \sum_i \mathbf{J}_i^T \mathbf{h}_i, \quad \mathbf{W} = \sum_i \mathbf{J}_i^T \mathbf{W}_i. \quad (87)$$

Thereby, the single update jobs of the summation loop are independent with respect to memory and computation; the final inserting into global memory space is the only critical task. According to Amdahl's law  $s_4 \leq 2.05$ ,  $e_4 \leq 0.5$  and according to Gustafson's law  $s_4 \leq 2.2$ ,  $e_4 \leq 0.55$  are expected for a four-core machine. The results in Fig. 12 show a program flow with parallelised updates, sequential summation, blocking of threads and not parallelised segments. A speed-up  $s_4 = 1.7$  has been achieved resulting in an efficiency factor  $e_4 = 0.43$ . So although, e.g. contact iterations cannot be simplified conceptually, the result of parallelisation is quite promising but there should still be possibilities to improve the scaling of the parallel parts about 0.07–0.12 efficiency values in comparison with the theory and to extend parallelisation to here not parallelised parts.

## 4 Conclusion

Computing time reduction is important especially for holistic validation. In the context of simplified CVT models the present paper detects possible bottlenecks and proposes solution possibilities. In particular, the determination of an initial state, stabilising methods and parallelisation is outlined. Summarising, a good initial state saves about 0.2 s real simulation time, implicit integration schemes propose a maximum realistic speed-up depending on the computer architecture and parallelisation according to Amdahl's law has a maximum asymptotic

speed-up of 3.23 for the discussed examples. Concerning the last two issues with a currently attained speed-up  $s_4 = 1.7$  it is worth to further improve and extend parallelisation techniques in future work.

## References

1. Acary V, Brogliato B (2008) Numerical methods for nonsmooth dynamical systems: applications in mechanics and electronics. Lecture notes in applied and computational mechanics, vol 35, 1st edn. Springer, Berlin
2. Bathe KJ, Bolourchi S (1979) Large displacement analysis of three-dimensional beam structures. *Int J Numer Meth Eng* 14:961–986
3. Brogliato B (1996) Nonsmooth impact mechanics: models, dynamics and control. Lecture notes in control and information sciences, vol 220, 1st edn. Springer, London
4. Brogliato B, ten Dam T, Paoli L, Genot F, Abadie M (2002) Numerical simulation of finite dimensional multibody nonsmooth mechanical systems. *Appl Mech Rev* 55:107–150
5. Chapman B, Gabriele, Jost, van der Pas R (2008) Using OpenMP: portable shared memory parallel programming. Scientific and engineering computation. MIT Press, Cambridge, MA
6. Crisfield MA (1990) A consistent co-rotational formulation for non-linear, three-dimensional, beam elements. *Comput Meth Appl Mech Eng* 81:131–150
7. von Dombrowski S, Schwertassek R (1999) Analysis of large flexible body deformation in multibody systems using absolute coordinates. In: Advances in computational multibody dynamics. Lisbon, 20th–23rd September 1999, Instituto Superior Tecnico. Lisbon, Portugal, pp 359–378
8. Flores P, Leine RI, Glocker C (2009) Modeling and analysis of rigid multibody systems with translational clearance joints based on the nonsmooth dynamics approach. In: Multibody dynamics 2009. Ecomas thematic conference. Warsaw, 29th June until 2nd July 2009
9. Förg M (2007) Mehrkörpersysteme mit mengenwertigen Kraftgesetzen: Theorie und Numerik, Fortschritt-Berichte VDI: Reihe 20, Rechnerunterstützte Verfahren, vol 411, als Manuskript gedruckt. VDI-Verl., Düsseldorf
10. Förg M (2009) MBSim – a multi body simulation software. GNU Lesser General Public License <http://mbsim.berlios.de>
11. Geier T (2007) Dynamics of push belt CVTs, Fortschrittberichte VDI: Reihe 12, Verkehrstechnik, Fahrzeugtechnik, vol 654, als Manuskript gedruckt. VDI-Verl., Düsseldorf
12. Gerbert G (1985) Influence of band friction on metal V-belt mechanics. Tech. Rep. 1985-08-25, Division of Machine Elements, Chalmers University of Technology, Göteborg
13. Gerstmayr J (2007) Absolute coordinate formulations for flexible multibody dynamics systems. PhD thesis, Johannes Kepler Universität Linz
14. Leine RI, Nijmeijer H (2006) Dynamics and bifurcations of non-smooth mechanical systems. Lecture notes in applied and computational mechanics, vol 18, Springer, Berlin
15. Leine RI, van de Wouw N (2008) Stability and convergence of mechanical systems with unilateral constraints. Lecture notes in applied and computational mechanics, vol 36. Springer, Berlin
16. Pfeiffer F (2008) Mechanical system dynamics. Lecture notes in applied and computational mechanics, vol 40, Springer, Berlin
17. Pfeiffer F, Glocker C (1996) Multibody dynamics with unilateral contacts, 1st edn. Wiley series in nonlinear science. Wiley, New York
18. Sattler H (1999) Stationäres Betriebsverhalten stufenlos verstellbarer Metallumschlingungsgetriebe. PhD thesis, Universität Hannover
19. Schindler T, Geier T, Ulbrich H, Pfeiffer F, van der Velde A, Brandsma A (2007) Dynamics of Pushbelt CVTs. In: Umschlingungsgetriebe: Ketten und Riemen – Konstruktion, Simulation und Anwendung von Komponenten und Systemen; Tagung Berlin, 21. und 22. Juni 2007, VDI-Verlag, Düsseldorf, VDI-Berichte; 1997

20. Schindler T, Ulbrich H, Pfeiffer F, van der Velde A, Brandsma A (2008) Pushbelt CVTs – A Nonsmooth Challenge. In: Proceedings of the 6th Euromech Nonlinear Dynamics Conference, St. Petersburg State University, St. Petersburg, Russia
21. Sedlmayr M (2003) Räumliche Dynamik von CVT-Keilkettengetrieben, Fortschrittberichte VDI: Reihe 12, Verkehrstechnik, Fahrzeugtechnik, vol 558, als Manuskript gedruckt. VDI-Verlag, Düsseldorf
22. Shabana A, Bauchau O, Hulbert G (2007) Integration of large deformation finite element and multibody system algorithms. *J Comput Nonlin Dynam* 2:351–359
23. Simo JC, Vu-Quoc L (1991) A geometrically-exact rod model incorporating shear and torsion-warping deformation. *Int J Solid Struct* 27:371–393
24. Wasfy T, Noor A (2003) Computational strategies for flexible multibody systems. *Appl Mech Rev* 56:553–613
25. Zander R (2009) Flexible multi-body systems with set-valued force laws, Fortschritt-Berichte VDI: Reihe 20, Rechnerunterstützte Verfahren, vol 420, als Manuskript gedruckt. VDI-Verlag, Düsseldorf, <http://mediatum2.ub.tum.de/node?id=654788>
26. Zander R, Schindler T, Friedrich M, Huber R, Förg M, Ulbrich H (2008) Non-smooth dynamics in academia and industry: recent work at TU München. *Acta Mech* 195:167–183



# Optimization-Based Design of Minimum Phase Underactuated Multibody Systems

Robert Seifried

**Abstract** An underactuated multibody system has less control inputs than degrees of freedom, e.g. due to passive joints or body flexibility. The analysis of the mechanical design of these kind of underactuated multibody systems might show that they are non-minimum phase, i.e. they have an internal dynamic which is not asymptotically stable. Therefore, feedback linearization is not possible, and also feed-forward control design for output trajectory tracking becomes a very challenging task. In this paper it is shown that through the use of an optimization procedure underactuated multibody systems can be designed in such a way that they are minimum phase. Thus feed-forward control design is significantly simplified and also feedback linearization of the underactuated multibody system is possible.

## 1 Introduction

Underactuated multibody systems possess less control inputs than degrees of freedom. Examples are multibody systems with body flexibility or passive joints. Due to the underactuation, the method of inverse dynamics known from fully actuated systems, see e.g. [2, 20], cannot be used for the design of controllers for end-effector trajectory tracking. Thus in the case of underactuation the controller design is much more involved than in the fully actuated case. Thereby, for trajectory tracking of underactuated multibody systems generally advanced modern nonlinear control techniques are necessary. Using concepts from differential geometric control theory, see [8, 14], the analysis of the mechanical design of underactuated multibody systems might show that they possess an internal dynamics. The internal dynamics under the constraint of constant zero output is called zero-dynamics. Systems with asymptotically stable zero-dynamics are called minimum phase, otherwise

---

R. Seifried (✉)

Institute of Engineering and Computational Mechanics, University of Stuttgart,  
Pfaffenwaldring 9, 70569 Stuttgart, Germany  
e-mail: [seifried@itm.uni-stuttgart.de](mailto:seifried@itm.uni-stuttgart.de)

non-minimum phase. For minimum phase systems feedback linearization, a nonlinear control technique well-suited for output trajectory tracking, is possible, see [4, 8, 14, 18]. Also the design of feed-forward control for minimum phase systems is rather straight forward. Conversely, for non-minimum phase systems feedback linearization is not possible. Also feed-forward control design is in this case much more complex and requires the numerical solution of a two-sided boundary value problem, see [6, 17, 21]. Therefore, the aim should be to design an underactuated multibody system in such a way that it is minimum phase. Thus already in the early state of the design process, mechanical design and control design should be considered concurrently. In [1] such a methodology is used to design differentially flat underactuated planar manipulators by using a special mass distribution. In this case, no internal dynamics remains and full-state linearization is possible; however the approach might require the use of larger counterweights.

In this paper, an optimization-based design procedure is proposed in order to design underactuated multibody systems in such a way that the internal dynamics becomes stable. In the proposed optimization procedure, the design parameters are the mass distribution of the multibody system. This can e.g. be achieved by additional masses which are added to defined locations of the multibody system. The optimization criteria is two-stage and firstly requires that all eigenvalues of the linearized zero-dynamics are in the left half-plane, and secondly that initial errors in the zero-dynamics decay rapidly. The analysis of this optimization problem shows that there are many local minima, and therefore a particle swarm optimization procedure is used. It is shown that minimum phase property can be achieved with only a modest increase of the total mass of the underactuated multibody system. The efficiency of this optimization-based design approach is demonstrated by simulations for manipulators with one and two degrees of underactuation, respectively.

## 2 Trajectory Tracking Control

Underactuated multibody systems with  $f$  degrees of freedom, generalized coordinates  $\mathbf{q} \in \mathbb{R}^f$  and inputs  $\mathbf{u} \in \mathbb{R}^m$  with  $m < f$ , i.e. control forces and torques, are considered. The nonlinear equation of motion is given by

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{B}(\mathbf{q})\mathbf{u}, \quad (1)$$

where  $\mathbf{M}$  is the mass matrix,  $\mathbf{k}$  the vector of generalized gyroscopic and centrifugal forces and  $\mathbf{g}$  the vector of applied forces. The input matrix  $\mathbf{B}$  distributes the control inputs  $\mathbf{u}$  onto the directions of the generalized coordinates. In the case of an underactuated multibody system the input matrix  $\mathbf{B}$  cannot be inverted and the classical approach of inverse dynamics cannot be used. Thus, more advanced nonlinear control techniques are necessary. In the following feedback linearization and feed-forward control design for output trajectory tracking of underactuated multibody systems is presented. These approaches are based on concepts from differential geometry and its theoretical background is described in [8, 14, 18].

## 2.1 Input–Output Normal-Form

The nonlinear input–output normal-form is the basis for feedback linearization as well as for feed-forward control design. This input–output normal-form is obtained by applying a coordinate transformation to the equation of motion. This diffeomorphic coordinate transformation is given by  $z = \Phi(x)$ , where  $x$  are the original coordinates and  $z$  are the coordinates of the input–output normal-form. A local diffeomorphic coordinate transformation exists if the Jacobian-matrix  $J = \frac{\partial \Phi(x)}{\partial x}$  is nonsingular. In general this transformation requires a state-space representation of the nonlinear system and the symbolic computation of Lie-derivatives of the output  $y$ . However, even for multibody systems with very few degrees of freedom, these symbolic calculations become very complicated. Therefore, in the following it is shown, that for a special type of system output  $y$  the nonlinear input–output normal-form can be directly derived from the second order differential equation of motion (1). In a first step the equation of motion (1) is partitioned into two parts,

$$\begin{bmatrix} M_{aa}(q) & M_{au}(q) \\ M_{au}^T(q) & M_{uu}(q) \end{bmatrix} \begin{bmatrix} \ddot{q}_a \\ \ddot{q}_u \end{bmatrix} + \begin{bmatrix} k_a(q, \dot{q}) \\ k_u(q, \dot{q}) \end{bmatrix} = \begin{bmatrix} g_a(q, \dot{q}) \\ g_u(q, \dot{q}) \end{bmatrix} + \begin{bmatrix} B_a(q) \\ B_u(q) \end{bmatrix} u. \quad (2)$$

Thereby the submatrix  $B_a \in \mathbb{R}^{m \times m}$  has rank  $m$ . The first  $m$  rows of the partitioned equation of motion (2) are referred to as actuated part associated with the  $m$  actuated coordinates  $q_a$ . The remaining  $f - m$  rows are referred to as the un-actuated part associated with the  $f - m$  un-actuated coordinates  $q_u$ . In the following, it is assumed that  $B_a = I$  is the identity matrix and  $B_u = 0$ . These special choices represent interesting cases of underactuated multibody systems in tree structure. Examples include rigid multibody systems with passive joints and planar elastic manipulators, where the shape functions of the elastic bodies are chosen according to clamped boundary conditions, see e.g. [5].

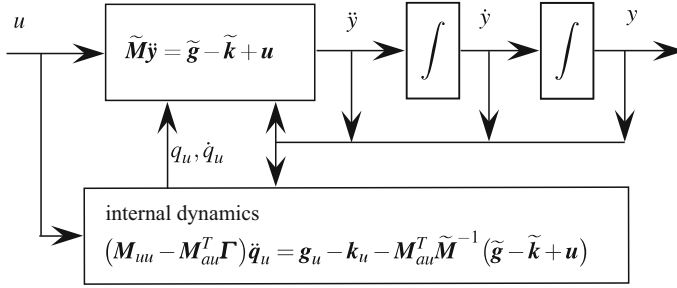
The nonlinear input–output normal-form depends on the choice of the system output  $y$ . Here it is assumed that the end-effector position can be approximately described by an output of form

$$y = q_a + \Gamma q_u, \quad (3)$$

where  $\Gamma \in \mathbb{R}^{m \times f_u}$ . This output is a linear combination of actuated and un-actuated generalized coordinates. For example such an output can be used to describe the end-effector position of elastic manipulators as shown in [5]. The partitioned equation of motion (2) is now transformed into the input–output normal-form with new coordinates  $y, q_u$ , see e.g. [17] for details. The input–output normal-form of the underactuated multibody system with the system output given by (3) reads

$$\widetilde{M} \ddot{y} = \widetilde{g} - \widetilde{k} + u, \quad (4)$$

$$[M_{uu} - M_{au}^T \Gamma] \ddot{q}_u = g_u - k_u - M_{au}^T \widetilde{M}^{-1} [\widetilde{g} - \widetilde{k} + u]. \quad (5)$$



**Fig. 1** Graphical representation of the input–output normal-form of underactuated MBS

In this nonlinear input–output normal-form, the terms are summarized according to the following convention:

$$\begin{aligned}\tilde{M} &= M_{aa} - (M_{au} - M_{aa}\Gamma)(M_{uu} - M_{au}^T\Gamma)^{-1}M_{au}^T, \\ \tilde{g} &= g_a - (M_{au} - M_{aa}\Gamma)(M_{uu} - M_{au}^T\Gamma)^{-1}g_u, \\ \tilde{k} &= k_a - (M_{au} - M_{aa}\Gamma)(M_{uu} - M_{au}^T\Gamma)^{-1}k_u.\end{aligned}$$

Equation (4) has dimension  $m$  and describes the relationship between the input  $u$  and output  $y$ . Since this is a second order differential equation for the output  $y$  the considered multibody systems have vector relative degree  $r = \{r_1, \dots, r_m\} = \{2, \dots, 2\}$ . The second part of the normal-form, given by (5) has dimension  $f - m$  and describes the so-called internal dynamics. A graphical representation of the nonlinear input–output normal-form of the underactuated multibody system with the system output given by (3) is shown in Fig. 1. The typical state-space representation of the input–output normal-form is omitted here, since analysis of the zero-dynamics, feedback linearization and feed-forward control design can be directly performed from the second order differential equations (4) and (5).

## 2.2 Analysis of the Internal Dynamics

The analysis of the stability of the internal dynamics is crucial for control design. Since this analysis is often quite complex the concept of zero-dynamics is used in drawing important conclusions about the stability of the internal dynamics. The zero-dynamics is the internal dynamics under the constraint that the output is kept identically zero, i.e.  $y = 0, \forall t$ . For the considered underactuated multibody systems the required control input for this task follows from (4) of the nonlinear input–output normal-form as

$$u_0 = \tilde{k}(0, q_u, 0, \dot{q}_u) - \tilde{g}(0, q_u, 0, \dot{q}_u). \quad (6)$$

Applying this input  $u_0$  to (5), the internal dynamics reduces to the zero-dynamics of the underactuated multibody system and reads

$$[M_{uu}(0, q_u) - M_{au}^T(0, q_u)\Gamma]\ddot{q}_u = g_u(0, q_u, 0, \dot{q}_u) - k_u(0, q_u, 0, \dot{q}_u). \quad (7)$$

As shown in [8, 14] a nonlinear system is called asymptotically (exponentially) minimum phase if the equilibrium point of the zero-dynamics is asymptotically (exponentially) stable. Otherwise, the system is called non-minimum phase. It should be noted, that the minimum phase property is independent of the choice of coordinates, and thus is invariant under a diffeomorphic coordinate transformation  $z = \Phi(x)$ . However, the minimum phase property depends on the system dynamics given by the equation of motion (1) and the choice of the system output  $y$ .

### 2.3 Feedback Linearization

Feedback linearization is based on the presentation of the nonlinear system in input–output normal-form. Then, the nonlinearities are cancelled using state-feedback, resulting in an exactly linearized system or subsystem, see [4, 8, 14, 18]. This approach is fundamentally different from Jacobian-linearization, in which the nonlinear system is approximated by a linear system. The nonlinearities in (4) can be cancelled by the linearizing feedback control law

$$u = \widetilde{M}(y, q_u)v + \widetilde{k}(y, q_u, \dot{y}, \dot{q}_u) - \widetilde{g}(y, q_u, \dot{y}, \dot{q}_u), \quad (8)$$

where  $v$  is a new input. It should be noted, that this linearizing feedback law depends on all states of the transformed system, i.e.  $y, \dot{y}, q_u, \dot{q}_u$ . Applying the linearizing feedback law (8) to the input–output normal-form (4) and (5) yields the input–output linearized system

$$\ddot{y} = v, \quad (9)$$

$$[M_{uu} - M_{au}^T\Gamma]\ddot{q}_u = g_u - k_u - M_{au}^T v. \quad (10)$$

The system consists of two subsystems. The first subsystem describes the linear relationship between the new input  $v$  and the output  $y$  and consists of  $m$  chains of two integrators. Therefore, this approach is also called input–output linearization. The second subsystem, resulting from the internal dynamics, is in general nonlinear. From (9) and (10) it is seen that only the first subsystem influences the output. Thus the feedback law (8) renders the states  $q_u, \dot{q}_u$  of the internal dynamics unobservable.

Since the first subsystem (9) is in canonical controllable form one can use linear control methods, such as eigenvalue assignment, to design with the new input  $v$  a feedback controller which influences the output  $y$  in a desired way. In this paper the control goal is trajectory tracking of a desired system output  $y = y_d$ , which is given by (3). Following [8, 14] the new input  $v$  in (9) might be used in order

to achieve asymptotic output trajectory tracking. The investigated underactuated multibody systems have vector relative degree  $r = \{r_1, \dots, r_m\} = \{2, \dots, 2\}$  and therefore the tracking control law reads

$$\mathbf{v} = \ddot{\mathbf{y}}_d + \mathbf{p}_1(\dot{\mathbf{y}}_d - \dot{\mathbf{y}}) + \mathbf{p}_0(\mathbf{y}_d - \mathbf{y}). \quad (11)$$

Thereby, the coefficients  $\mathbf{p}_0$ ,  $\mathbf{p}_1$  are diagonal matrices. In the special case of a constant reference trajectory,  $\mathbf{y} = \mathbf{0}$ ,  $\forall t$ , the tracking control law (11) reduces to a control law for stabilization around a stationary point. Introducing the output trajectory error  $\mathbf{e} = \mathbf{y}_d - \mathbf{y}$  and applying control law (11) to the linearized subsystem (9) yields the linear error dynamics

$$\ddot{\mathbf{e}} + \mathbf{p}_1\dot{\mathbf{e}} + \mathbf{p}_0\mathbf{e} = \mathbf{0}. \quad (12)$$

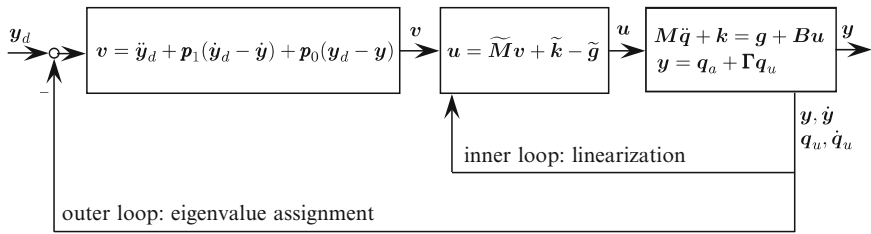
From this follows that the diagonal matrices  $\mathbf{p}_0$ ,  $\mathbf{p}_1$  can be used to place the eigenvalues of the error dynamics in the left half-plane. Then, due to a suitable choice of  $\mathbf{p}_0$ ,  $\mathbf{p}_1$  the system output converges to the desired reference trajectory. However, the task of controller design is not only to influence the output in a desired way, but also to achieve that the whole dynamics of the system behaves well. Thus, the control design given by control law (11) is only valid, if the unobservable states  $\mathbf{q}_u$ ,  $\dot{\mathbf{q}}_u$  of the internal dynamics remain bounded.

An inspection of the internal dynamics (10) under the control law (11) shows, that it can be viewed as a nonlinear time-varying system driven by the desired output trajectory  $\mathbf{y}_d$ . First of all, the special case of tracking a constant output  $\mathbf{y} = \mathbf{0}$  has to be considered. In this case the tracking control law (11) coincides with a stabilizing control law and leads to the requirement that the zero-dynamics has to be asymptotically stable, i.e. the nonlinear system is minimum phase, see [8, 14, 18]. In the case of trajectory tracking, this is initially a rather weak condition, however from a practical point of view it is the crucial point for the analysis of the behavior of the internal dynamics. For a non-constant desired output trajectory  $\mathbf{y}_d$  additional conditions exist which strengthen the requirement of minimum phase, see [8, 14, 18] for details. For example in [18] it is shown that for exponentially minimum phase nonlinear systems the desired trajectory  $\mathbf{y}_d$  and its first  $r_i - 1$  derivatives must be small enough in order to guarantee that control law (11) yields to convergence of the tracking error  $\mathbf{e}$  and bounded internal states  $\mathbf{q}_u$ ,  $\dot{\mathbf{q}}_u$ .

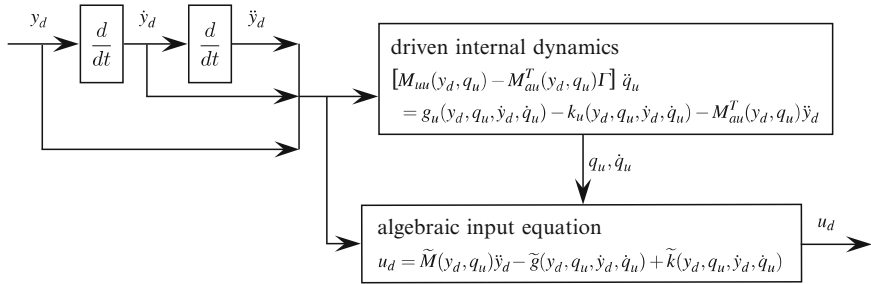
The presented control structure consists of an inner and an outer loop, and is shown schematically in Fig. 2. In the inner loop exact linearization is achieved by using state-feedback law (8). The outer loop is used for eigenvalue assignment of the error dynamics (12) by control law (11).

## 2.4 Feed-forward Control Design

The feed-forward control design is based on an inverse model which provides the input  $\mathbf{u}_d$  required for exact reproduction of a desired output trajectory  $\mathbf{y} = \mathbf{y}_d$ .



**Fig. 2** Control structure with inner and outer loop



**Fig. 3** Graphical representation of feed-forward control of an underactuated MBS

In order to account for small disturbances and uncertainties the feed-forward control has to be supplemented by additional feedback control. This yields a so-called control structure with two design degrees of freedom. Both parts of this control system can be designed largely independent from each other. The input  $u_d$  computed by the feed-forward control follows from (4) as

$$u_d = \tilde{M}(y_d, q_u) \ddot{y}_d - \tilde{q}(y_d, q_u, \dot{y}_d, \dot{q}_u) + \tilde{k}(y_d, q_u, \dot{y}_d, \dot{q}_u). \quad (13)$$

The computation of the input  $u_d$  depends on the desired output  $y_d, \dot{y}_d$  and the un-actuated states  $q_u, \dot{q}_u$ . These latter ones are the solution of the internal dynamics (5) which is driven by  $y_d, \dot{y}_d$  and  $u_d$ . Replacing  $u_d$  in the internal dynamics (5) by (13) yields for the un-actuated states  $q_u, \dot{q}_u$  the differential equation

$$\begin{aligned} & [M_{uu}(y_d, q_u) - M_{au}^T(y_d, q_u) \Gamma] \ddot{q}_u \\ & = g_u(y_d, q_u, \dot{y}_d, \dot{q}_u) - k_u(y_d, q_u, \dot{y}_d, \dot{q}_u) - M_{au}^T(y_d, q_u) \ddot{y}_d. \end{aligned} \quad (14)$$

In summary, the inverse model consists of three parts which are shown schematically in Fig. 3. The first part represents a chain of two differentiators for the desired output vector  $y_d$ , producing the values  $\dot{y}_d$  and  $\ddot{y}_d$ . The second part of the inverse model is the driven internal dynamics (14) for the  $q_u$  coordinates. The third part of the inverse model is the algebraic equation (13) which computes from these values the desired input  $u_d$ .

Several methods for model inversion exist which differ in the solution of the internal dynamics (14). In classical model inversion [7] the  $\mathbf{q}_u, \dot{\mathbf{q}}_u$  variables are found through forward integration of the internal dynamic (14) from the starting time point  $t_0$  to the final time point  $t_f$ , using the initial values  $\mathbf{q}_u(t_0) = \mathbf{q}_{u_0}$ ,  $\dot{\mathbf{q}}_u(t_0) = \dot{\mathbf{q}}_{u_0}$ . However, in order to use the input  $\mathbf{u}_d$  in a feed-forward control, it must be bounded. Thus depending on the stability of internal dynamics forward integration of the internal dynamics might yield unbounded  $\mathbf{q}_u, \dot{\mathbf{q}}_u$  values and thus unbounded inputs  $\mathbf{u}_d$ . Therefore, classical inversion can only be used for feed-forward control design if the internal dynamics (14) remains bounded, which implies that only minimum phase systems can be treated. In the case of non-minimum phase systems a bounded feed-forward control can be computed by stable inversion as described in [6, 17, 21]. However, in this approach the internal dynamics (14) is solved as a two-sided boundary value problem. This yields a non-causal solution, i.e. pre- and post-actuation is necessary. In general the solution of this boundary value problem must be pre-computed for each desired output trajectory, e.g. by using finite differences as proposed in [21].

An alternative to the presented approach for feed-forward control design is proposed in [3] for flat mechanical systems. Thereby the inversion problem is solved by deriving a set of differential–algebraic equations which is then solved numerically. A similar approach is used in [12] for minimum phase flexible manipulators.

### 3 Design of Stable Zero-Dynamics

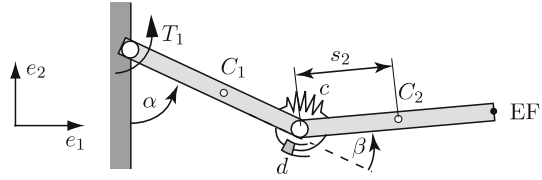
The analysis of the initial design of an underactuated multibody system might show that it possess an unstable zero-dynamics. Due to the previously discussed shortcomings and difficulties in trajectory control of non-minimum phase systems, it is desired to design the multibody system in such a way that the zero-dynamics is stable. Then feedback linearizable is possible and also the design of feed-forward control is significantly simplified. As shown in Sect. 2.2, the zero-dynamics depends on the choice of the system output  $\mathbf{y}$  and the equation of motion of the multibody system. Output relocation is a method where a different system output  $\mathbf{y}$  is chosen in order to achieve minimum phase property. However, the use of this approach is limited if trajectory tracking of an end-effector point is aspired. Thus, minimum phase property can only be achieved by modifying the system dynamics, which means the mechanical design of the underactuated multibody system must be altered.

#### 3.1 Identification of Possible Design Parameters

In the following physical parameters which influence the stability properties of the zero-dynamics are identified. For this exemplary investigation, a single rotational arm is considered which consists out of two links connected by one active and one



**Fig. 4** Rotational arm with one active and one passive joint



passive joint. The rotational arm is shown in Fig. 4, whereby the center of mass of link  $i = 1, 2$  is denoted by  $C_i$ . The links have length  $l_i$ , mass  $m_i$ , inertia  $I_{i_z}$  and the position of the center of mass is described by  $s_i$ . An input torque  $T_1$  acts on link 1. Link 2 is connected by a passive joint to link 1, which is supported by a spring-damper combination with spring constant  $c$  and damping coefficient  $d$ . The arm is described by the generalized coordinates  $\alpha$  and  $\beta$ , whereby  $\beta$  denotes the un-actuated coordinate. The arm moves perpendicular to the direction of gravity.

For this investigation the system output of the rotational arm is given by the linear combination  $y = \alpha + \Gamma\beta$ . In this case the zero-dynamics of the rotational arm reads

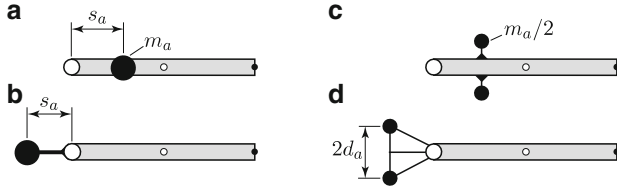
$$[(1 - \Gamma)(I_{2_z} + m_2 s_2^2) - l_1 m_2 s_2 \Gamma \cos \beta] \ddot{\beta} = -c\dot{\beta} - d\dot{\beta} - l_1 m_2 s_2 \Gamma^2 \beta^2 \sin \beta. \quad (15)$$

This shows that the zero-dynamics of the rotational arm is influenced by the length  $l_1$  of the first link, the value  $\Gamma$  of the output, the coefficients  $c, d$  of the spring-damper combination and the mass distribution of the second link, given by its mass  $m_2$ , inertia  $I_{2_z}$  and center of mass  $s_2$ . For a further analysis the linearized zero-dynamics around the equilibrium point  $\beta = 0$  is considered,

$$[(1 - \Gamma)(I_{2_z} + m_2 s_2^2) - l_1 m_2 s_2 \Gamma] \ddot{\beta} + d\dot{\beta} + c\beta = a_2 \ddot{\beta} + a_1 \dot{\beta} + a_0 \beta = 0, \quad (16)$$

where  $a_2, a_1, a_0$  correspond to the coefficients of the characteristic polynomial. Thus, the linearized zero-dynamics of the rotational arm is only asymptotically stable if all coefficients  $a_2, a_1, a_0$  have the same sign and are non-zero, see e.g. [13]. Since the constants  $c, d$  of the spring-damper combination are by nature positive also the coefficient  $a_2$  has to be positive. Thus, in this case  $c, d$  can only be used to shape the dynamic response of the zero-dynamics, but cannot be used to change its stability property. The factor  $\Gamma$  is assumed to be fixed in order to give a suitable approximation of the end-effector position. Also in the following it is assumed that the arm topology is fixed, i.e. the length of the links cannot be changed. Consequently only the mass distribution of the second link remains as design variable to alter the stability properties of the zero-dynamics of the rotational arm. This analysis is representative for the systems considered in the following.

The mass distribution of the un-actuated link, given by its mass  $m_2$ , inertia  $I_{2_z}$  and center of mass  $s_2$ , could be used directly as design variables. However, these quantities are coupled, and the optimization might yield values which cannot be realized from an engineering point of view. Therefore, in the following approach a basic



**Fig. 5** Four possible design variants for altering the mass distribution of a link

homogeneous initial design of the underactuated multibody system is assumed. Then, in the case of a non-minimum phase initial design an additional small balancing mass  $m_a$  is added to each un-actuated link in order to alter the stability property of the zero-dynamics, thus achieving minimum phase behavior. The additional mass  $m_a$  is added to the location  $s_a$  on the link which changes the center of mass of the combined body. To gain a higher flexibility the additional mass  $m_a$  might be added as counterweight to the link. An additional increase of the inertia  $I_{2z}$  of the un-actuated link can be achieved if the mass  $m_a$  is located by an offset  $d_a$  away from the axis of the link. Four possible design variants for an un-actuated link are shown schematically in Fig. 5.

In summary, the three design variables  $\mathbf{p} = [m_a, s_a, d_a]$  can be used to alter the mass distribution of an un-actuated link and yield

$$\begin{aligned} m_2 &= \bar{m}_2 + m_a, \quad s_2 = \frac{\bar{m}_2 \bar{s}_2 + m_a s_a}{m_2}, \\ I_{2z} &= \bar{I}_{2z} + \bar{m}_2 (\bar{s}_2 - s_2)^2 + m_a (s_a - s_2)^2 + m_a d_a^2. \end{aligned} \quad (17)$$

Thereby  $\bar{m}_2, \bar{I}_{2z}, \bar{s}_2$  denote the values of the initial design of the un-actuated link. In order to obtain a viable physical design, bounds have to be put on the design variables which results in the feasible design space for one un-actuated link

$$P = \{\mathbf{p} \in \mathbb{R}^3 \mid 0 \leq m_a \leq m_{a_{\max}}, s_{a_{\min}} \leq s_a \leq s_{a_{\max}}, 0 \leq d_a \leq d_{a_{\max}}\}. \quad (18)$$

For underactuated multibody systems with several passive joints, the design variables  $\mathbf{p}$  and the feasible design space  $P$  are the collection of the design variables of all un-actuated links.

### 3.2 Optimization Criteria

For this design process an optimization procedure is proposed using the previously identified design parameters to modify the mass distribution of the un-actuated bodies. The primary design goal is to achieve a stable zero-dynamics, such that

the underactuated multibody system is feedback linearizable. The zero-dynamics is given by (7) and depends only on the un-actuated states  $\mathbf{q}_u, \dot{\mathbf{q}}_u$  and the design variables  $\mathbf{p}$ . Therefor, the zero-dynamics can be written as

$$[\mathbf{M}_{uu}(\mathbf{p}, \mathbf{q}_u) - \mathbf{M}_{au}^T(\mathbf{p}, \mathbf{q}_u)\boldsymbol{\Gamma}]\ddot{\mathbf{q}}_u = \mathbf{g}_u(\mathbf{p}, \mathbf{q}_u, \dot{\mathbf{q}}_u) - \mathbf{k}_u(\mathbf{p}, \mathbf{q}_u, \dot{\mathbf{q}}_u), \quad (19)$$

and the system matrix of the linearized zero-dynamics (19) is denoted by  $\mathbf{A}(\mathbf{p})$ .

In order to obtain a powerful mechanical design, not only minimum phase behavior must be guaranteed, but also additional goals must be achieved by the design process. Firstly, the design should be robust to uncertainties in the mass distribution of the un-actuated bodies. This mean, the system should remain its minimum phase property even if in the physical construction there are small unknown variations. These can be either in the initial mass distribution or in the optimal design variables. Secondly, the zero-dynamics should not only be stable, but also disturbances should decay rapidly. This is especially important in order to avoid that disturbances yield large undesired vibrations of the internal dynamics during trajectory tracking. Therefore, a two-stage computation of the optimization criteria  $f(\mathbf{p})$  is proposed, which should be minimized in the course of the optimization:

1. In the first step, Lyapunov's indirect method is used, see [11]. It requires that all eigenvalues of the linearized zero-dynamics are in the left half-plane,

$$\text{Re}[\lambda(\mathbf{A}(\mathbf{p}))] < 0. \quad (20)$$

In order to achieve robustness against uncertainty in the mass distribution of the un-actuated bodies it is also desired that the eigenvalues for several perturbation parameter sets  $\mathbf{p} + \Delta\mathbf{p}$  are in the left half-plane,

$$\text{Re}[\lambda(\mathbf{A}(\mathbf{p} + \Delta\mathbf{p}))] < 0. \quad (21)$$

In this paper 16 designs with perturbation are tested, whereby the design parameters are varied in different combination by up to 5% around the nominal values  $\mathbf{p}$ . Thus a safety region is created around the nominal design  $\mathbf{p}$ . If at least one eigenvalue of the nominal or the designs with perturbation has a non-negative real part, a large default value for the optimization criteria  $f(\mathbf{p})$  is returned. Otherwise, the linearized analysis shows asymptotic stability and it is proceeded with step 2. It should be noted, that her only a point-wise robustness test is performed. In order to guarantee robustness over the entire region of uncertainties this point-wise test can be replaced by a  $\mu$ -analysis, see e.g. [19] for details on analysis of system robustness.

2. If all eigenvalues are in the left half-plane, the final optimization criteria  $f(\mathbf{p})$  is calculated. In order to achieve good damping properties it is required that initial errors in the nonlinear zero-dynamics (19) decay rapidly. The disturbance is given by the initial conditions  $\mathbf{q}_u(t_0) = \mathbf{q}_{u0}, \dot{\mathbf{q}}_u(t_0) = \dot{\mathbf{q}}_{u0}$ . The optimization

criteria  $f(\mathbf{p})$  is then described by the cumulated squared error of the  $f - m$  un-actuated coordinates  $\mathbf{q}_u$  in respect to the equilibrium point  $\mathbf{q}_u = \mathbf{0}$  of the zero-dynamics. This is given by,

$$f(\mathbf{p}) = \sum_{i=1}^{f-m} \int_{t_0}^{t_1} q_{u_i}^2 dt, \quad (22)$$

where  $t_1$  describes the final time of the simulation. Besides evaluating the damping properties of the zero-dynamics, this second step in the criteria computation provides also a very good indication about the behavior of the nonlinear zero-dynamics. It gives an indication if the zero-dynamics remains stable in the case that the internal states are pushed by a disturbance further away from the equilibrium point. In order to achieve a good trade-off between the damping property and avoiding large mass increases the optimization criteria (22) can be extended in the sense of weighted criteria to

$$f(\mathbf{p}) = \sum_{i=1}^{f-m} \int_{t_0}^{t_1} q_{u_i}^2 dt + w \sum_{i=1}^{f-m} m_{a_i}, \quad (23)$$

where  $w$  is a weighting factor.

### 3.3 Particle Swarm Optimization

In the optimization procedure the criteria function  $f(\mathbf{p})$  should be minimized with respect to the design variables  $\mathbf{p}$ . Due to the two-stage criteria computation, the optimization problem is discontinuous. Also an analysis of this optimization problem shows, that there are many local minima, often surrounded by areas of instability of the zero-dynamics. The complexity of the topology of the optimization criteria increases with the number of passive joints. Therefore, gradient based optimization algorithms cannot be used and stochastic optimization algorithms must be deployed. Here a particle swarm optimization procedure is used. This is a population based optimization method which originates in the study and simulation of social behavior of bird and fish flocks, see [9]. The basic idea is the modeling of social interaction between individual particles of a population on the quest for the best point in the feasible design space. Thereby, it is aspired to use the collective intelligence of a swarm to solve complex optimization problems. A detailed analysis of swarm intelligence is given in [10].

As advantages of the particle swarm optimization it should be named that no gradient information is necessary, the solution is independent of initial sets of design parameters  $\mathbf{p}$ , and there are no requirements on smoothness or continuity of the optimization criteria. This approach is well suited for finding global minima and

is often easy to program and to adjust to specific problems. The used algorithm is a Matlab implementation presented in [15, 16], and has been already successfully applied in the optimization of multibody systems.

Compared to gradient based methods, a general disadvantage of stochastic optimization algorithms is their large computational expense due to a large amount of criteria evaluations. In the criteria computation the far most time-consuming part is the time-integration of the zero-dynamics in the second stage. However, in the first stage of the criteria computation many unfeasible designs are filtered out and thus the number of time integrations is heavily reduced by the restriction on locally stable designs. In summary, the two-stage type of criteria calculation in combination with particle swarm optimization is an efficient way to design a stable and robust zero-dynamics which also shows good damping properties.

## 4 Application Examples

In this section the efficiency of the optimization-based design approach for underactuated multibody systems is presented using simulations of two application examples. These two examples are planar underactuated manipulators with a kinematic redundancy and one and two passive joints, respectively. In the first example the advantages of the presented optimization-based design and control approach is compared to other possible control strategies and its robustness is demonstrated. The second example shows different designs which can be achieved in an efficient way by this optimization-based design approach.

### 4.1 Manipulator with One Passive Joint

The first investigated underactuated manipulator has one passive joint and is shown schematically in Fig. 6. The manipulator moves along the horizontal plane

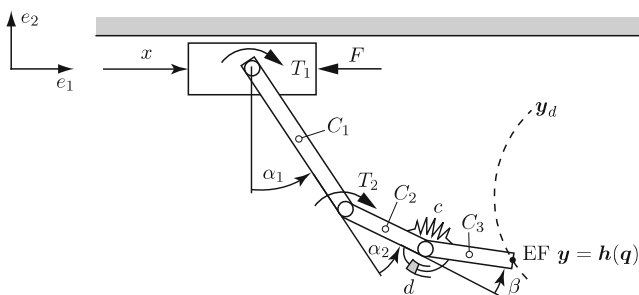


Fig. 6 Underactuated manipulator with one passive joint

**Table 1** Initial parameters for underactuated manipulator with one passive joint

Cart	$m_c = 3kg$		
Arm 1	$m_1 = 6.875kg$	$I_1 = 0.5743 \frac{kg}{m^2}$	$l_1 = 1.0m$
Arm 2	$m_2 = 3.438kg$	$I_2 = 0.0723 \frac{kg}{m^2}$	$l_2 = 0.5m$
Arm 3	$m_3 = 3.438kg$	$I_3 = 0.0723 \frac{kg}{m^2}$	$l_3 = 0.5m$
	$c = 50 \frac{Nm}{rad}$	$d = 0.25 \frac{Nm s}{rad}$	

and consists of a cart on which a chain of three arms is mounted. The homogenous arms have length  $l_1$  and  $l_2 = l_3$ . The manipulator is described by the generalized coordinate  $\mathbf{q} = (x, \alpha_1, \alpha_2, \beta)^T$  and is actuated by the control input  $\mathbf{u} = (F, T_1, T_2)^T$ . The third arm is connected by a passive joint to arm 2 which is supported by a parallel spring-damper combination with spring constant  $c$  and damping coefficient  $d$ . The initial physical parameters of the manipulator are summarized in Table 1.

The control goal is to force the end effector to follow a predefined trajectory as closely as possible. For a somewhat stiff spring-damper combination, the angle  $\beta$  remains small. Then, the end-effector position can be approximated by

$$\mathbf{r}_{EF} = \begin{bmatrix} x + l_1 \sin(\alpha_1) + l_2 \sin(\alpha_1 + \alpha_2) + l_3 \sin(\alpha_1 + \alpha_2 + \beta) \\ -l_1 \cos(\alpha_1) - l_2 \cos(\alpha_1 + \alpha_2) - l_3 \cos(\alpha_1 + \alpha_2 + \beta) \end{bmatrix} \quad (24)$$

$$\approx \begin{bmatrix} x + l_1 \sin(\alpha_1) + (l_2 + l_3) \sin(\alpha_1 + \alpha_2 + \Gamma\beta) \\ -l_1 \cos(\alpha_1) - (l_2 + l_3) \cos(\alpha_1 + \alpha_2 + \Gamma\beta) \end{bmatrix}, \quad (25)$$

which can be described by the linearly combined output

$$\mathbf{y} = (y_1, y_2, y_3)^T = (x, \alpha_1, \alpha_2 + \Gamma\beta)^T. \quad (26)$$

From geometrical consideration it turns out that the factor  $\Gamma = \frac{l_2}{l_1 + l_2} = 0.5$  provides a good approximation of the position of the end-effector point as long as  $\beta$  remains small. This choice of output is motivated from control of elastic manipulators, see [5]. However, it should be noted, that due to this approximation a small tracking error for the end-effector position has to be expected.

The analysis of the zero-dynamics of this manipulator with output  $\mathbf{y}$  shows that it is identical to the zero-dynamics of the rotational arm presented in Sect. 3.1 and is given by (15). With the physical parameters given in Table 1 the system is non-minimum phase. In order to achieve feedback linearizability the presented optimization procedure is used. As discussed in Sect. 3.1 the design parameters are an additional mass  $m_a$  and its location  $s_a$  and offset  $d_a$  which is added to the un-actuated arm 3. The bounds of the optimization parameters are summarized in Table 2 and are chosen in such a way that a viable mechanical design can be achieved. The optimization criteria given by (23) with  $t_1 = 0.25s$  and  $w = 0.75 \cdot 10^{-4}$  is chosen. The result of this optimization is also given in Table 2. The optimization result shows, that the required additional mass is only  $0.531kg$ ,

**Table 2** Optimization parameters-bounds and optimization result

Parameters	$m_a$ (kg)	$s_a$ (m)	$d_a$ (m)
Lower bound	0	-0.333	0
Upper bound	1	0.333	0.167
Opt. result	0.531	-0.333	0.167

which is a small increase of 3.1% compared to the total mass of the initial design. The additional mass is mounted with an offset as counterweight to the un-actuated link, corresponding to design variant **d** shown in Fig. 5. The computed values for distance  $s_a$  and offset  $d_a$  are on the bounds of the design space. Thus, a further improvement might be achieved by increasing these bounds.

#### 4.1.1 System Without Disturbances and Uncertainties

The obtained optimal design is tested considering a half-circular end-effector trajectory. The center of the half-circle is at position  $(0, -1.5\text{ m})$  and the radius is  $1\text{ m}$ . The end-effector point should follow the trajectory in the short time period of  $1.5\text{ s}$ , which describes an aggressive manoeuvre. Also the kinematic redundancy should be used to perform a secondary task, which is moving the cart from starting position  $-1\text{ m}$  to the final position  $1\text{ m}$ .

The simulation results of the end-effector trajectory for the manipulator with optimized mass distribution in combination with feedback linearization is presented in Fig. 7. These results are compared to those using two alternative control concepts applied to the initial design of the manipulator. The first alternative control approach is a stable inversion based feed-forward control of the non-minimum phase initial system combined with a time-variant Linear Quadratic Regulator (LQR). The stable inversion yields a pre-and post-actuation phase and its computation is described for this manipulator in [17]. The second alternative control approach is a feedback linearization, whereby minimum phase property of the initial system is achieved by output relocation. An analysis of the zero-dynamics shows that in this example output relocation requires  $\Gamma \leq 0.4$ . In the presented example  $\Gamma = 0.4$  is chosen.

The simulations show, that the best results are achieved by the design with optimized mass distribution of the un-actuated arm. Hereby the errors of the end-effector point trajectory are the smallest of the three approaches. In this case the maximum error of the end-effector trajectory is  $2.1\text{ mm}$ . This is better than the results achieved with the stable inversion approach, which yields a maximum error of about  $3.2\text{ mm}$ . The output relocation approach yields the worst performance with a maximum end-effector trajectory error of  $28\text{ mm}$ . Since in all three cases there are no disturbances the output  $y$  is tracked exactly. The end-effector point errors occur due to the approximation of the end-effector point by (25). It turns out, that due to the changed mass distribution of the un-actuated arm in the optimized design, the  $\beta$  coordinate remains smaller than in the case of the stable inversion of the initial system. Thus the approximation given by (25) is even better, and yields as a side effect a smaller trajectory error for the optimized design. The output relocation yields an

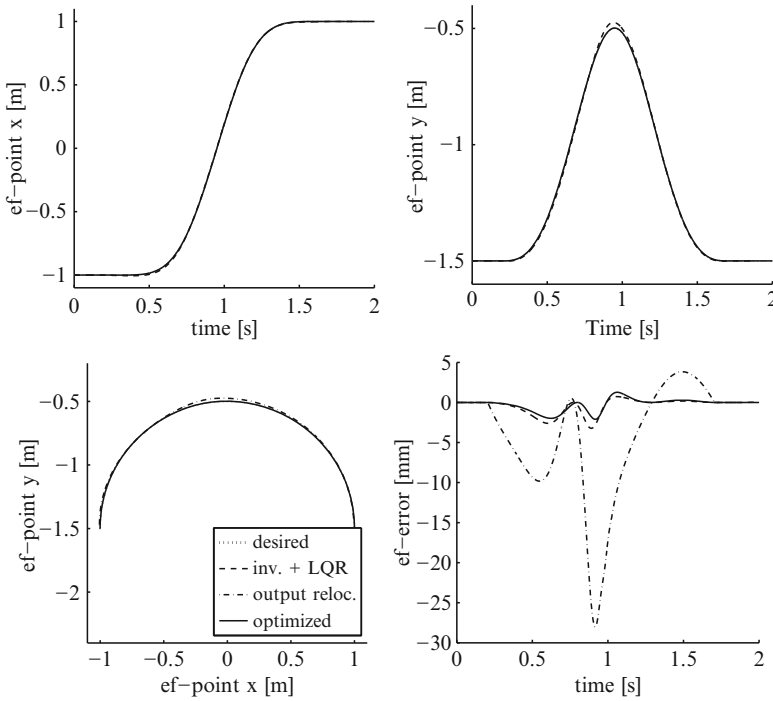
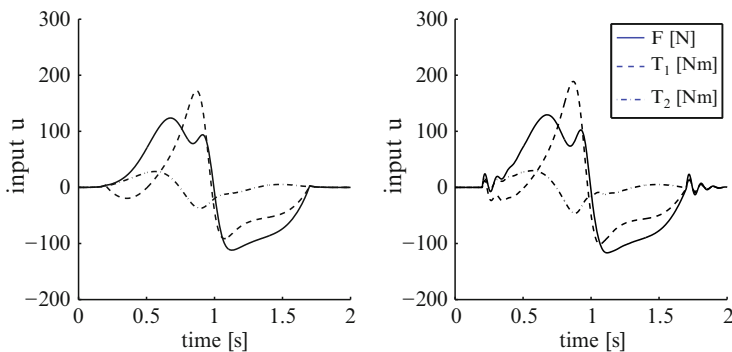


Fig. 7 End-effector point trajectory for manipulator with one passive joint

unsatisfying performance since with  $\Gamma = 0.4$  only an insufficient approximation of the end-effector point can be achieved. This approach might be useful in the case of stabilization but not in the case of trajectory tracking and is therefore not further considered.

The control inputs for the feedback linearization of the optimized manipulator and for the stable inversion of the initial design are presented in Fig. 8. A small mass increase occurs due to the additional mass in the optimization. Thus, the energy consumption when tracking this trajectory increases by approximately 5% compared to the stable inversion of the initial system. However, feedback linearization has several significant advantages compared to stable inversion. Feedback linearization yields an algebraic control law which is relatively easy to implement and independent of the desired output trajectory. In contrast, stable inversion of a non-minimum phase system has to be computed off-line for each desired trajectory separately by the numerical solution of a two-sided boundary value problem, see e.g. [6, 17, 21]. Also the time-variant LQR requires the numerical solution of the differential Riccati equation which also has to be performed off-line for each output trajectory. Finally, it has to be noted, that stable inversion of a non-minimum phase system yields a so-called pre-actuation phase. In the case of a kinematic redundancy, this pre-actuation phase might be avoidable as proposed in [17], however this goes along with an increase of the consumed energy.





**Fig. 8** Control inputs for stable inversion + LQR (*left*) and feedback linearization of optimized system (*right*)

#### 4.1.2 System Under Disturbances and Uncertainties

The results presented in the previous section were obtained by simulation of an ideal system. However, in reality there are always disturbances and uncertainties which might deteriorate the performance of the control strategy. Therefore, simulations are performed where the total mass, inertia and the center of mass of the un-actuated arm is increased by 5%, the stiffness of the spring is increased by 15% and the damping is set to  $d = 0$ . Also measurement noise is considered, which is added as white noise to the generalized coordinates. The simulation results for the end-effector trajectory using the feedback linearization of the optimized minimum phase system and for stable inversion with LQR of the initial non-minimum phase design are shown in Fig. 9. The corresponding control inputs are shown in Fig. 10.

The results show, that both strategies still work very well under these uncertainties and disturbances. However, the maximum trajectory tracking error using the feedback linearization is with  $3.9\text{ mm}$  about 42% smaller than in the case of stable inversion which is  $6.7\text{ mm}$ . The weighting matrices of the LQR are chosen in such a way that under these specific parameter uncertainties a good performance is achieved and the noise in the control inputs is in the same magnitude as in case of feedback linearization, see Fig. 10. This shows, that the proposed optimization-based design procedure yields a robust design in which the internal dynamics remains stable under these uncertainties and disturbances, and thus feedback linearization yields satisfying results.

## 4.2 Manipulator with Two Passive Joints

The second investigated underactuated manipulator is shown schematically in Fig. 11. It is similar to the first example, however it has two passive joints and an additional load is added to the end-effector point. The actuated generalized

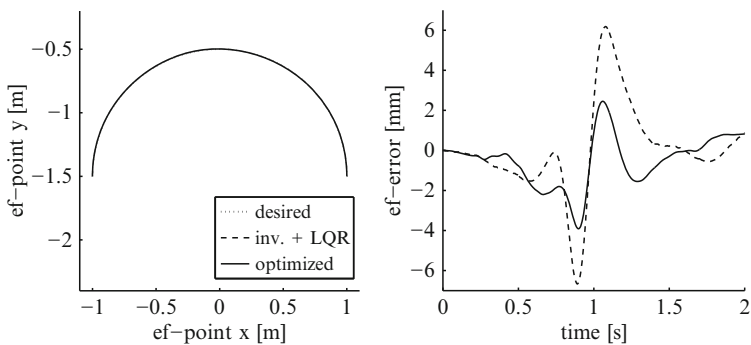


Fig. 9 End-effector point trajectory under disturbances for manipulator with one passive joint

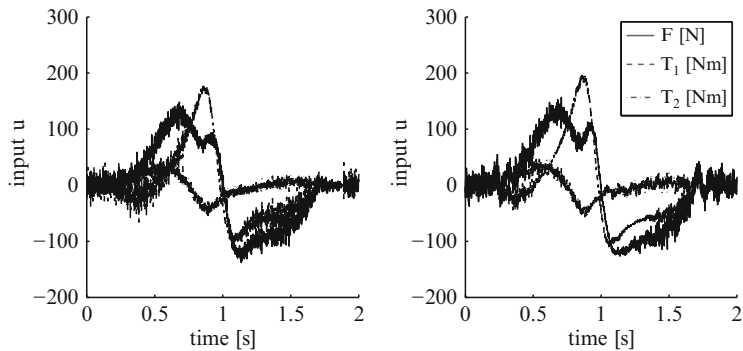


Fig. 10 Control inputs for stable inversion + LQR (left) and feedback linearization of optimized system (right) under disturbances

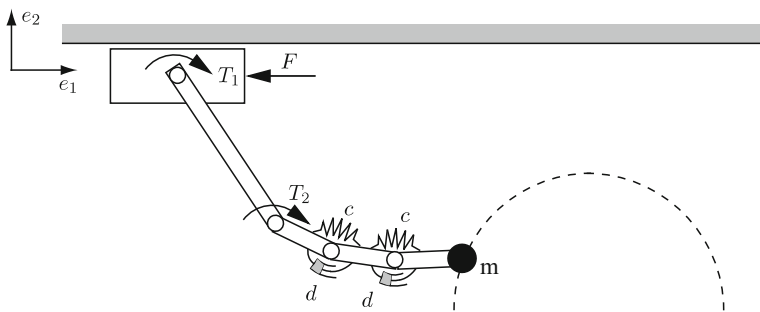


Fig. 11 Underactuated manipulator with two passive joints and end-point load

**Table 3** Initial parameters for underactuated manipulator with two passive joints

Cart	$m_c = 3 \text{ kg}$			
Arm 1	$m_1 = 6.875 \text{ kg}$	$I_1 = 0.5743 \frac{\text{kg}}{\text{m}^3}$	$l_1 = 1.0 \text{ m}$	
Arm 2	$m_2 = 2.292 \text{ kg}$	$I_2 = 0.0217 \frac{\text{kg}}{\text{m}^3}$	$l_2 = 0.333 \text{ m}$	
Arm 3	$m_3 = 2.292 \text{ kg}$	$I_3 = 0.0217 \frac{\text{kg}}{\text{m}^3}$	$l_3 = 0.333 \text{ m}$	
Arm 4	$m_4 = 2.292 \text{ kg}$	$I_4 = 0.0217 \frac{\text{kg}}{\text{m}^3}$	$l_4 = 0.333 \text{ m}$	
Load	$m_l = 6 \text{ kg}$	$I_l = 0.0147 \frac{\text{kg}}{\text{m}^3}$		
	$c = 400 \frac{\text{Nm}}{\text{rad}}$	$d = 0.25 \frac{\text{Nms}}{\text{rad}}$		

coordinates are  $\mathbf{q}_a = (x, \alpha_1, \alpha_2)^T$  and the un-actuated generalized coordinates are  $\mathbf{q}_u = (\beta_1, \beta_2)^T$ . The physical properties are summarized in Table 3. In order to approximate the end-effector point a linearly combined system output is chosen as

$$\mathbf{y} = \mathbf{q}_a + \mathbf{\Gamma} \mathbf{q}_u \quad \text{with} \quad \mathbf{\Gamma} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \Gamma_1 & \Gamma_2 \end{bmatrix}. \quad (27)$$

In this example a good approximation is achieved for  $\Gamma_1 = 2/3$  and  $\Gamma_2 = 1/3$ . However, this yields a non-minimum phase initial design, and the proposed optimization procedure is used to design a minimum phase manipulator.

An analysis of the zero-dynamics of this manipulator shows, that the optimization problem is more complex than in the first example. Many local minima with similar criteria value exist and the influence of the mass distribution of the two un-actuated arms on the zero-dynamics is coupled. From a practical point of view the primary goal is to find in an efficient way a viable design. This does not have to be necessarily the global optima, but a local optima with very good performance is sufficient. Therefore three cases are considered using different bounds and optimization criteria. For each case 10 optimization runs are performed which are terminated after 120 s.

Table 4 shows for Case 1 the bounds on the parameters and the results using the optimization criteria given by (22). Case 2 uses the optimization criteria given by (23) and the weighting factor  $w = 1.5 \times 10^{-4}$ . The bounds and results for Case 2 are summarized in Table 5. Table 6 presents the results of Case 3 where the criteria given by (22) is used, however no offset  $d_a$  is allowed. For each parameter set the value of the optimization criteria  $f$  is given in the tables. Also the maximum trajectory error  $e_{max}$  and the increase  $\Delta E$  of the energy consumption is given in comparison to the solution of the stable inversion approach of the initial non-minimum phase design for the previously presented half-circular trajectory. Thereby the trajectory error  $e_{max}$  and the increase of consumed energy  $\Delta E$  are computed from simulations of the optimized designs.

**Table 4** Case 1 – bounds and optimization results

	$m_{a_1}$ (kg)	$s_{a_1}$ (m)	$d_{a_1}$ (m)	$m_{a_2}$ (kg)	$s_{a_2}$ (m)	$d_{a_2}$ (m)	$f$ ( $10^{-4}$ )	$e_{max}$ (mm)	$\Delta E$ (%)
Lo. bound	0	−0.333	0	0	−0.333	0			
Up. bound	1	0.333	0.167	2	0.333	0.167			
Run 1	0.627	−0.128	0.130	2.000	−0.333	0.167	1.31	2.388	11.4
Run 2	0.997	−0.094	0.086	1.999	−0.333	0.167	1.28	2.375	13.1
Run 3	0.665	−0.148	0.000	1.999	−0.333	0.167	1.27	2.370	11.6
Run 4	0.707	−0.139	0.042	1.989	−0.333	0.080	1.68	2.465	12.6
Run 5	0.535	−0.175	0.000	1.999	−0.333	0.167	1.28	2.376	11.0
Run 6	0.200	−0.325	0.167	2.000	−0.333	0.167	1.35	2.406	9.5
Run 7	0.905	−0.111	0.055	1.994	−0.333	0.167	1.27	2.371	12.6
Run 8	0.625	−0.148	0.066	1.999	−0.333	0.167	1.28	2.375	11.3
Run 9	0.731	−0.130	0.069	1.999	−0.333	0.167	1.28	2.374	11.9
Run 10	0.900	−0.088	0.125	1.986	−0.333	0.005	1.91	2.364	13.3

**Table 5** Case 2 – bounds and optimization results

	$m_{a_1}$ (kg)	$s_{a_1}$ (m)	$d_{a_1}$ (m)	$m_{a_2}$ (kg)	$s_{a_2}$ (m)	$d_{a_2}$ (m)	$f$ ( $10^{-4}$ )	$e_{max}$ (mm)	$\Delta E$ (%)
Lo. bound	0	−0.333	0	0	−0.333	0			
Up. bound	1	0.333	0.167	2	0.333	0.167			
Run 1	0.408	−0.195	0.119	1.573	−0.333	0.167	4.67	2.410	9.1
Run 2	0.203	−0.333	0.118	1.635	−0.333	0.166	4.41	2.416	8.5
Run 3	0.218	−0.327	0.066	1.610	−0.333	0.167	4.42	2.410	8.4
Run 4	0.287	−0.262	0.117	1.896	−0.333	0.034	5.34	2.360	9.9
Run 5	0.228	−0.319	0.067	1.665	−0.333	0.167	4.44	2.445	8.8
Run 6	0.342	−0.210	0.159	1.576	−0.333	0.167	4.59	2.415	8.8
Run 7	0.364	−0.207	0.137	1.578	−0.333	0.167	4.61	2.411	8.9
Run 8	0.218	−0.327	0.061	1.606	−0.333	0.167	4.40	2.409	8.4
Run 9	0.210	−0.333	0.080	1.787	−0.333	0.162	4.53	2.547	9.5
Run 10	0.241	−0.294	0.120	1.643	−0.333	0.166	4.45	2.420	8.7

The results of Case 1–3 given in Tables 4–6 show, that after optimization runs of only 120 s a viable design for the manipulator with two passive joints is found. It must be mentioned, that only about 1 in 10 simulation runs do not yield a non-minimum phase design within 120 s, whereby these runs are not accounted for in the presented tables. This shows, that the presented approach is a very time-efficient method to design minimum phase underactuated systems. For each of the three cases, most of the presented optimizations yield similar parameter sets. However, there are some variations in the different designs, indicating local minima.

It is worth to notice, that the additional masses are always placed as counterweights, whereas the mass added to the second un-actuated arm is significantly larger than the one added to the first un-actuated arm. The most energy-efficient

**Table 6** Case 3 – bounds and optimization results

	$m_{a1}$ (kg)	$s_{a1}$ (m)	$d_{a1}$ (m)	$m_{a2}$ (kg)	$s_{a2}$ (m)	$d_{a2}$ (m)	$f$ ( $10^{-4}$ )	$e_{max}$ (mm)	$\Delta E$ (%)
Lo. bound	0	-0.333	0	0	-0.333	0			
Up. bound	1	0.333	0	2	0.333	0			
Run 1	0.993	-0.108	0.000	2.000	-0.333	0.000	1.79	2.386	13.8
Run 2	0.802	-0.128	0.000	1.998	-0.333	0.000	1.81	2.383	12.9
Run 3	0.999	-0.107	0.000	1.998	-0.333	0.000	1.79	2.387	13.9
Run 4	0.862	-0.121	0.000	2.000	-0.333	0.000	1.80	2.385	13.3
Run 5	0.839	-0.123	0.000	1.999	-0.333	0.000	1.80	2.384	13.2
Run 6	0.727	-0.138	0.000	2.000	-0.333	0.000	1.81	2.383	12.6
Run 7	0.845	-0.123	0.000	1.999	-0.333	0.000	1.81	2.384	13.2
Run 8	0.814	-0.126	0.000	2.000	-0.333	0.000	1.80	2.384	13.1
Run 9	0.717	-0.140	0.000	1.998	-0.333	0.000	1.82	2.382	12.6
Run 10	0.505	-0.182	0.000	1.998	-0.333	0.000	1.85	2.377	11.6

designs are obtained by Case 2. This is due to the fact that in the criteria computation also the added mass is considered. In this case an average of 8.9% of additional energy is required for tracking the half-circular trajectory. Case 1 which does not account for the additional mass in the criteria computation yield designs with larger mass increase. An average of 11.8% of additional energy is required in order to tracking the half-circular trajectory. Evaluating the squared error of the un-actuated coordinates as given by (22) shows, that in Case 1 the damping property of the zero-dynamics is better than in Case 2. Both, Cases 1 and 2 yield an offset  $d_a$ . If no offset is allowed, the added masses have to be larger, as shown by Case 3. This yields an increase of the consumed energy by about 13.1% for the half-circular trajectory. Also the damping property of the zero-dynamics is worse than in Case 1, as indicated by the criteria value  $f$ . The damping property is comparable to the ones in Case 2. Thus, the use of an offset  $d_a$  improves the performance significantly.

## 5 Conclusions

A design approach for minimum phase underactuated multibody systems is presented. This design approach is based on the optimization of the zero-dynamics, and yields a stable and robust internal dynamics with good damping properties. Then, for trajectory tracking feedback linearization is possible, and also feed-forward control design is significantly simplified. As design parameters the mass distributions of the un-actuated bodies are identified. The mass distribution can be influenced by small masses which are added to the un-actuated bodies. For the optimization a particle swarm algorithm is used. Due to the two-stage optimization criteria calculation the optimization procedure is very time-efficient and yields reliable results. The efficiency of the approach is demonstrated by simulation using two planar underactuated manipulators with one and two passive joints, respectively. Only a small

increase of the total mass is necessary to achieve a feedback linearizable system. In the future a more sophisticated design parametrization is aspired to achieve minimum phase underactuated multibody systems without an increase of the total mass.

## References

1. Agrawal S, Sangwan V (2008) Differentially flat designs of underactuated open-chain planar robots. *IEEE T Robot* 24:1445–1451
2. Ast A, Eberhard P (2006) Flatness-based control of parallel kinematics using multibody systems - simulation and experimental results. *Arch Appl Mech* 76:181–197
3. Blajer W, Kolodziejczyk K (2004) A geometric approach to solving problems of control constraints: theory and a DAE framework. *Multibody Syst Dyn* 11:343–364
4. Brockett R (1978) Feedback invariants for nonlinear systems. In: *Proceedings of the 7th IFAC world congress*, Helsinki, Finland
5. De Luca A (1998) Trajectory control of flexible manipulators. In: Siciliano B, Valavanis K (eds) *Control problems in robotics and automation*. Springer, London, pp 83–104
6. Devasia S, Chen D, Paden B (1996) Nonlinear inversion-based output tracking. *IEEE T Automat Contr* 41:930–942
7. Hirschorn R (1979) Invertibility of multivariable nonlinear control systems. *IEEE T Automat Contr* 24:855–865
8. Isidori A (1995) *Nonlinear control systems*. Springer, London
9. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of the international conference on neural networks*, Perth, Australia, pp 1942–1948
10. Kennedy J, Eberhart R (2001) *Swarm intelligence*. Morgan Kaufmann, San Francisco
11. Khalil H (2002) *Nonlinear systems*. Prentice Hall, Upper Saddle River
12. Moberg S, Hanssen S (2007) A DAE approach to feedforward control of flexible manipulators. In: *Proceedings of the 2007 IEEE international conference on robotics and automation*, pp 3439–3444
13. Müller P C, Schiehlen W (1985) *Linear vibrations*. Martinus Nijhoff Publishers, Dordrecht
14. Sastry S (1999) *Nonlinear systems: analysis, stability and control*. Springer, New York
15. Sedlaczek K (2007) *Zur Topologieoptimierung von Mechanismen und Mehrkörpersystemen* (in German). Schriften aus dem Institut für Technische und Numerische Mechanik der Universität Stuttgart, Shaker Verlag, Aachen
16. Sedlaczek K, Eberhard P (2006) Using augmented Lagrangian particle swarm optimization for constrained problems in engineering. *Struct Multidiscip O* 32:277–286
17. Seifried R, Eberhard P (2009) Design of feed-forward control for underactuated multibody systems with kinematic redundancy. In: Ulbrich H, Ginzinger L (eds) *Motion and vibration control: selected papers from MOVIC 2008*. Springer, pp 275–284
18. Soltine J-J, Li W (1991) *Applied nonlinear control*. Prentice Hall, Englewood Cliffs
19. Skogestad S, Postlethwaite R (2005) *Multivariable feedback control*. Wiley, Chichester
20. Spong M, Hutchinson S, Vidyasagar M (2006) *Robot modeling and control*. Wiley, Hoboken
21. Taylor D, Li S (2002) Stable inversion of continuous-time nonlinear systems by finite-difference methods. *IEEE T Automat Contr* 47:537–542

# GPU-Based Parallel Computing for the Simulation of Complex Multibody Systems with Unilateral and Bilateral Constraints: An Overview

Alessandro Tasora, Dan Negrut, and Mihai Anitescu

**Abstract** This work reports on advances in large-scale multibody dynamics simulation facilitated by the use of the Graphics Processing Unit (GPU). A description of the GPU execution model along with its memory spaces is provided to illustrate its potential parallel scientific computing. The equations of motion associated with the dynamics of large system of rigid bodies are introduced and a solution method is presented. The solution method is designed to map well on the parallel hardware, which is demonstrated by an order of magnitude reductions in simulation time for large systems that concern the dynamics of granular material. One of the salient attributes of the solution method is its linear scaling with the dimension of the problem. This is due to efficient algorithms that handle in linear time both the collision detection and the solution of the nonlinear complementarity problem associated with the proposed approach. The current implementation supports the simulation of systems with more than one million bodies on commodity desktops. Efforts are under way to extend this number to hundreds of millions of bodies on small affordable clusters.

## 1 Introduction

Gauging through simulation the mobility of tracked and/or wheeled vehicles on granular terrain (sand and/or gravel) for commercial (construction equipment industry), military (off-road mobility), and deep space exploration (Rover mobility

---

A. Tasora (✉)  
University of Parma, Parma, Italy  
e-mail: [tasora@ied.unipr.it](mailto:tasora@ied.unipr.it)

D. Negrut  
University of Wisconsin, Madison, WI-53706, USA  
e-mail: [negrut@wisc.edu](mailto:negrut@wisc.edu)

M. Anitescu  
Argonne National Laboratory, Argonne, IL-60439, USA  
e-mail: [anitescu@mcs.anl.gov](mailto:anitescu@mcs.anl.gov)

on Martian terrain) applications leads to very challenging multibody dynamics problems. In the past, when applicable, the only feasible approach to these and other granular dynamics dominated problems was to approximate the discrete nature of the material with a continuum representation. For the classes of problems of interest here, such as material mixing, vehicle mobility on sand, piling up of granular bulk material, the flow in pebble bed nuclear reactors, rate of flow in silos, stability of brick buildings to earthquakes, etc., a continuum representation of the problem is either inadequate or paints with too wide of a brush the dynamics of interest. Tackling head on the discrete problem, characterized by a large number of bodies that interact through frictional contact and might have vastly different mass/inertia attributes, has not been feasible in the past.

The computational multibody dynamics landscape has experienced recently changes fueled by both external and internal factors. In terms of the former, sequential computing appears to lose momentum at a time when the microprocessor industry ushers in commodity many-core hardware. In terms of internal factors, contributions made in understanding and handling frictional contact [1–11], have led to robust numerical algorithms that can tackle sizeable granular dynamics problems. This paper discusses how the interplay of these two factors will enable in the near future a discrete approach to investigating the dynamics of systems with hundreds of millions of rigid bodies.

The paper is organized as follows. Section 2 starts with a brief discussion of three roadblocks that adversely impact the potential of sequential computing and limit its future role in computational science in general, and computational multibody dynamics in particular. An argument is made that in large scale multibody dynamics emphasis should be placed on implementations that can leverage commodity high performance parallel computing. In this context, an overview is presented of NVIDIA's hardware architecture, which is adopted herein when tackling large scale multibody dynamics problems. The discussion focuses on a description of the parallel execution model, execution scheduling, and memory layout. Section 3 details how large scale frictional contact problems associated with granular dynamics are solved by a computational approach that maps well onto parallel execution hardware available on the GPU. The approach implemented has two compute intensive parts: the solution of a cone complementarity problem (CCP) and the resolution of a collision detection (CD) analysis. In both cases, the solution embraced draws on parallel computing and a discussion of the CCP algorithm adopted concludes Section 3. Section 4 demonstrates the use of the solution approach implemented. First, the paper briefly reports on the largest granular dynamics problems solved using the methodology discussed in Section 3. Next, a pebble bed nuclear reactor flow problem compares the efficiency of the parallel implementation on the GPU to that of the sequential implementation. The paper closes with concluding remarks and a discussion of future directions of research.



## 2 Review of Computing on the Graphics Processing Unit

As pointed out in [12], three road blocks prevent traditional sequential computing from experiencing future major gains in flop rate: the memory block, the instruction level parallelism block, and the power dissipation block. The first one is a consequence of the fact that as the data processing power of a CPU core increases, the number of memory transactions in the time unit also goes up. From 1986 to 2000, CPU speed improved at an annual rate of 55% while memory access speed only improved at a 10% rate. One outcome of this trend was an increase in the likelihood of cache misses, which have been partially alleviated by employing hyper-threading technologies and considering ever increasing cache memories. Nonetheless, cache misses occur and they lead to the CPU waiting for chunks of data moved over a 32.5 GB/s connection that currently connects the CPU to the RAM. The second block stems from the exhaustion of the idea of speculative execution of future instructions to produce results ahead of time and make them available to the processor in case the actual computational path was correctly anticipated. However, this speculative execution strategy necessitates power and is plagued by a combinatorial increase in the number of possible computational paths. This translates into a short future execution horizon that can be sampled by these techniques. The attractive attribute of this strategy is that the programmer doesn't have to do anything to speed up the code. Instead, the CPU takes upon itself the task of employing this strategy. On the flip side, this avenue of speeding up execution has been thoroughly taken advantage of and its potential has been already fulfilled. Thirdly, the amount of power dissipated by a CPU/unit area has approached that of a nuclear plant [13]. Since the power dissipated is proportional to the square of the microprocessor clock frequency, it becomes apparent that significant microprocessor frequency increases, which were primarily responsible for past reductions in computational times in commodity scientific computing, are a thing of the past.

One bright spot in this bleak background against which the future of commodity hardware for scientific computing is projected comes from the consensus in the microprocessor industry that for at least one more decade Moore's law will hold. The law states that the number of transistors that can be placed inexpensively on an integrated circuit is doubling approximately every 2 years. Since this translates into a steady increase in the number of microprocessors that can be packed on the unit area, Moore's law indirectly defines the source of future increases in flop rate in scientific computing. Specifically, rather than hoping for frequency gains, one will have to count on an increase in number of cores as the means for speeding up simulation.

Figure 1 confirms this trend by comparing top flop rates for the CPU and GPU. Since the plot compares double precision (DP) CPU flop rates with single precision (SP) rates for the GPU, the relevant point is not made by the absolute values. Rather, the trends are more important: the slope for the CPU is significantly smaller than that of the GPU. Table 1 partially explains the momentum behind parallel computing on the GPU. The last generation of NVIDIA cards packs 1.4 billion transistors, reaching 3 billion with the release of Fermi in early 2010, to produce a GPU with

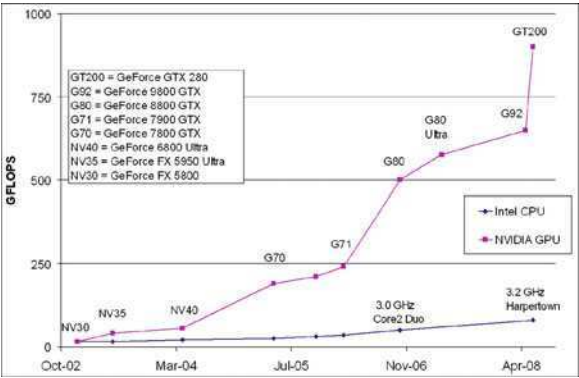


Fig. 1 Evolution of flop rate, comparison CPU vs. GPU

**Table 1** CPU vs. GPU comparison. Flop rates reported are in single precision (SP) for the GPU and double precision (DP) for the CPU

	Tesla C1060	Intel I7 975 Extreme
Cores	240	4
Memory	4 GB	32 KB L1 cache/core 256 KB L2 cache/core 8 MB L3 for all cores
Clock	1.33 GHz	3.20 GHz
Bandwidth	102 GB/s	32.0 GB/s
FLOPS	933 × 10 <sup>9</sup> (SP)	70 × 10 <sup>9</sup> (DP)

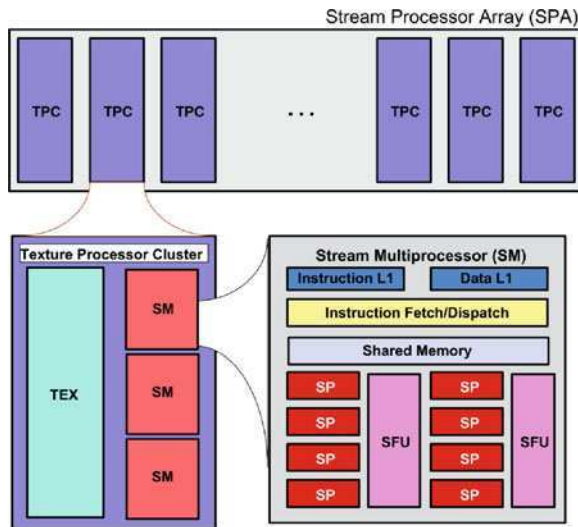
240 scalar processors, or 512 on Fermi. Their clock frequency is lower, that is, 1.33GHz, thus partially alleviating the heat dissipation issue. Yet, the GPU compensates through a larger memory bandwidth (likely to increase to more than 200 GB/s on Fermi) and sheer number of scalar processors.

The idea of using the graphics card for scientific computing dates back more than one decade. Their use was motivated by the sheer amount of computational power available on the GPU. Fueled by a steady demand for a more realistic video game experience, the GPU experienced a continuous increase in flop rate to facilitate the rendering of more realistic visual effects at a rate of 20 frames/s or higher. The original graphics pipeline operated through graphics shaders and was meant to perform the same set of operations on multiple data sets. The data here is the information associated with a pixel; the operations were the set of instructions necessary to determine the state of each pixel of the screen. At high resolutions, this required a large number of threads to process in parallel the information that would make possible the output of one frame. This computational model, in which one set of instructions is applied for many instances of data, is called SIMD (single instruction multiple data). It is the paradigm behind processing data on the GPU and was leveraged before 2006 by drawing on existing graphics application programming interfaces (API) such as OpenGL and DirectX.

However, scientific computing through a graphics API was both cumbersome and rigid. It was cumbersome since any data processing task had to be cast into a shader operation. This either required a lot of imagination, or outright prevented one from using GPU computing for more complicated tasks. The approach was also rigid in that it only allowed a limited number of memory transaction operations (for instance one thread could only write to one memory location), it lacked certain arithmetic operations (such as integer and bit operations), and implementation of the IEEE754 standard for arithmetic operations was of secondary importance.

The GPU computation landscape was revolutionized by the release in 2006 of the version 1.0 of the CUDA Software Development Kit (SDK) and library [14], which eliminated the vast majority of the barriers that prevented the use of the GPU for scientific computing. CUDA allows the user to write “C with extensions” code to directly tap into the computational resources of the GPU through a run-time API. The CPU, typically called the host, is linked to the GPU, called the device, through a Peripheral Component Interconnect Express 2.0 (PCIe 2.0  $\times$  16) connection. This connection supports an 8.0 GB/s data transfer rate and represents the conduit for data exchange between the host and device.

The hardware layout of the latest generation of NVIDIA graphics cards for scientific computation called Tesla is schematically shown in Fig. 2. The GPU is regarded as one big Stream Processor Array (SPA) that for the Tesla C1060 hosts a collection of 10 Texture Processor Clusters (TPC). Each TPC is made of a texture block (called TEX in Fig. 2), and more importantly, of three Stream Multiprocessors (SM). The SM, sometimes also called the multiprocessor, is the quantum of scalability for the GPU hardware. Thus, entry level graphics cards might have four SMs, such as is



**Fig. 2** Hardware layout for the Tesla C1060 card. The SPA has ten TPCs, each with three SMs, each of which has eight SPs for a total of 240 SPs

the case for GPUs like NVIDIA's 9700M GT which are used in computer laptops. High end GPUs, such as the NVIDIA GTX 280, have 30 SMs. The Tesla C1060 has also 30 SMs since the SPA has ten TPCs, each with three SMs. Finally, each SM has eight Scalar Processors (SP). It is these SPs that eventually execute the instructions associated with each function that is processed on the GPU. Specifically, the device acts as a co-processor for the host, which sends down to the device tasks for parallel execution. For this computational model to be effective, at least two requirements must be met. First, the ratio of arithmetic operations to data transfer should be high enough to cover the transfer overhead associated with the 8.0 GB/s data transfer from host to device for processing, and then back to the host for subsequent use. Second, the task sent for completion on the GPU, encapsulated in a C function called kernel, should have a high level of fine grain SIMD type parallelism.

For effective use of the available SMs, a kernel function must typically be executed by a number of threads in excess of 30,000. In fact, the more threads are launched, the larger the chance of full utilization of the GPU's resources. It should be pointed out that there is no contradiction in 240 SPs being expected to process hundreds of thousands or millions of parallel invocations of a kernel function. In practice, the largest number of times a kernel can be asked to be executed on Tesla C1060 is more than two trillion ( $65,535 \times 65,535 \times 512$ ) times.

When discussing about running kernels on the GPU, it is important to make a distinction between being able to execute a kernel function a large number times, and having these executions run in parallel. In practice, provisions should be made that there are enough instances of the kernel function that are lined up for execution so that the 240 SPs never become idle. This explains the speed-ups reported in conjunction with GPU computing when applications in image processing, quantum chemistry, and finance have run up to 150 times faster on the GPU although the peak flop rate is less than 10 times higher when compared to the CPU. For the latter, cache misses place the CPU in idle mode waiting for the completion of a RAM transaction. Conversely, when launching a job on the GPU that calls for a very large number of executions of a kernel function, chances are that the scheduler will always find warps, or collections of threads, that are ready for execution. In this context, the SM scheduler is able to identify and park with almost zero overhead the warps that wait for memory transaction completion and quickly feed the SM with warps that are ready for execution. The SM scheduler (which manages the "Instruction Fetch/Dispatch" block in Fig. 2) can keep tabs on a pool of up to 32 warps of threads, where each warp is a collection of 32 threads that are effectively executed *in parallel*. Thus, for each SM, the scheduler jumps around with very little overhead in an attempt to find, out of the 32 active warps, the next warp ready for execution. This effectively hides memory access latency.

Note that the number of *threads* that are executed in parallel (32 of them), is typically orders of magnitude smaller than the number of times the kernel function will be executed by a user specified number of threads. The latter can be specified through a so called execution configuration, which is an argument passed along to the GPU with each kernel function call. The execution configuration is defined by specifying the number of blocks of threads that the user desires to launch.

The maximum number of blocks is  $65,535 \times 65,535$ ; i.e., one can specify a two dimensional grid of blocks. Additionally, one has to indicate the number of threads that each block will be made up of. There is an upper limit of threads in a block, which currently is set to 512. When invoking an execution configuration, that is a grid of  $m$  blocks each with  $n$  of threads, the kernel function that is invoked to be executed on the device will be executed a number of  $m \times n$  times. In terms of scheduling, the  $m$  blocks are assigned to the available SMs and therefore a high end GPU comes ahead since the  $m$  blocks will end up assigned to four SMs on an entry level GPU, or to 30 SMs on a high end GPU. The assignment of blocks to SMs might lead to the simultaneous execution of more than one block/SM. Yet, this number cannot be larger than eight, which is more than sufficient since when they land on the same SM the eight blocks of threads are supposed to share resources. Indeed, due to the limited number of registers and amount of shared memory available on a SM, a sharing of resources between many threads ( $n \times$  the number of blocks executed on the SM) makes very unlikely the scenario of having a large number of blocks simultaneously running on one SM.

In terms of block scheduling, as one block of threads finishes the execution of the kernel function on a certain SM, another block of threads waiting to execute is assigned to the SM. Consequently, the device should be able to do scheduling at two levels. The first is associated with the assignment of a block to an SM that is ready to accept a new block for execution. What simplifies the scheduling here is the lack of time slicing associated with block execution: if a block is accepted for execution on an SM, no other block is accepted by that SM before it finishes the execution of a block that it is already dealing with. The second level of scheduling, which is more challenging, has to do with the scheduling for execution of one of the potentially 32 warps of threads that each SM can handle at any given time. Note that all the 32 threads in one warp execute the same instruction, even though this means, like in the case of if-then-else statements, serializing the code of the if-branches and running no-ops for certain threads in the warp (this thread divergence adversely impacts overall performance and should be avoided whenever possible). However, when switching between different warps, the SM typically executes different instructions when handling different warps; in other words, time slicing is present in thread execution.

In conclusion, one Tesla C1060, can be delegated with the execution of a kernel function up to approximately 2 trillion times. However, at each time, since there are 30 SMs available in this card, it will actively execute at most  $30,720 = 30 \times 32$  warps  $\times$  32 threads at any time. Moreover, as shown in Fig. 3, existing motherboards can accommodate up to four Tesla C1060 cards, which effectively supports up to  $122,880 = 4 \times 30,720$  threads being active at the same time. The single precision flop rate of this setup is approximately 3,600 billion operations/s.

It was alluded before that one of the factors that prevent an SM from actually running at full potential; i.e., managing simultaneously 32 warps of threads, is the exhaustion of shared memory and/or register resources. Each SM has 16 KB of shared memory in addition to 16,384 four byte registers. If the execution of the kernel function requires a large amount of either shared memory or registers, it is



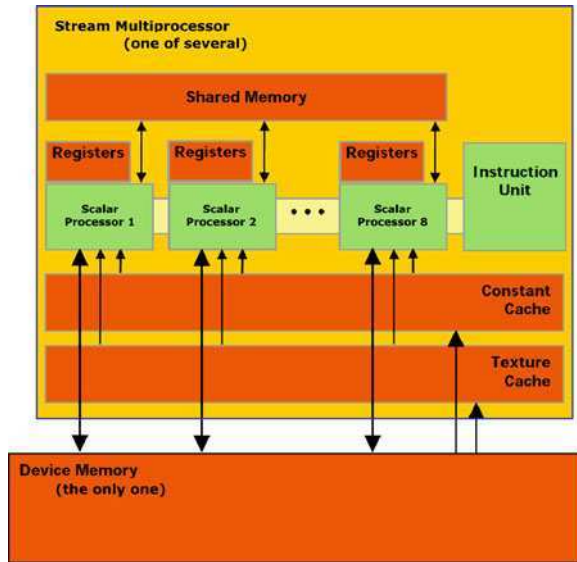
**Fig. 3** Image of GPU and desktop with a set of four cards that can be controlled by one CPU. There is no direct memory access between the four GPUs. The HW configuration in the figure is as follows. Processor: AMD Phenom II X4 940 Black Edition. Power supply 1: Silverstone OP1000-E 1000W. Power supply 2: Corsair CMPSU-750TX 750W. Memory: G.SKILL 16GB (4× 4 GB) 240-Pin DDR2. Case: LIAN LI PC-P80 ATX Full Tower. Motherboard: Foxconn Destroyer NVIDIA nForce 780a SLI. HDD: Western Digital Caviar Black 1TB 7200 RPM 3.0 Gb/s. HSF: Stock AMD. Graphics: 4x NVIDIA Tesla C1060

clear that the SM does not have enough memory available to host too many threads executing the considered kernel. Consequently, the ability of the SM to hide global memory access latencies with arithmetic instructions decreases since there are less warps that it can switch between.

In addition to shared memory and registers, as shown in Fig. 4, each thread has access to global memory (4 GB of it on a Tesla C1060), constant memory (64 KB), and texture memory, the latter in an amount that is somewhat configurable but close to the amount of constant memory. Additionally, there is so called local memory used to store data that is not lucky enough to occupy a register and ends up in the global memory (register overflow). Effectively, local memory is virtual memory that is carved out of the global memory and, in spite of the word “local”, it is associated with high latency. In this context, accessing data in registers has practically no latency, shared memory transactions have less than four clock cycles of latency, as do cached constant and texture memory accesses. Global memory transactions are never cached and, just like un-cached constant and texture memory accesses or accesses to local memory, they incur latencies of the order of 600 clock cycles. Note that typically the device does not have direct access to host memory. There are ways to circumvent this by using mapped page-locked memory transactions, but this is an advanced feature not discussed here.

For the GPU to assist through co-processing a job run on the CPU, the host must first allocate memory and move data through the PCI connection into the device memory (global, texture, or constant memory spaces). Subsequently, a kernel function is launched on the GPU to process data that resides in the device memory. At that point, blocks of threads executing the kernel function access data stored in device memory. In unsophisticated kernels they can immediately process the data; alternatively, in more sophisticated kernel functions, they can use the

**Fig. 4** GPU memory layout. Device memory refers to the combination of global, texture, and constant memory spaces. Arrows indicate the way data can move between different memory spaces and SM. While the device memory is available to threads running on any SM, the registers, shared memory, and cached constant and texture data is specific to each SM



shared memory and registers to store the data locally and thus avoid costly device memory accesses. If avoiding repeated data transfers between host and device is the first most important rule for effective GPU computing, avoiding repeated high-latency calls to device memory is the second most important rule to be observed in GPU computing. It should be pointed out that device memory access can be made even more costly when the access is not structured (uncoalesced). Using CUDA terminology, the device memory accesses result in multiple transactions if the data accessed by a warp of threads is scattered rather than nicely coalesced (contiguous) in memory. For more details, the interested reader is referred to [14].

One common strategy for avoiding race conditions in parallel computing is the synchronization of the execution at various points of the code. In CUDA, synchronization is possible but with a caveat. Specifically, threads that execute the kernel function yet belong to different blocks cannot be synchronized. This is a consequence of the earlier observation that there is no time slicing involved in block execution. When there are thousands of blocks that are lined up for execution waiting for their turn on one of the 30 SM of a Tesla C1060, it is clear that there can be no synchronization between a thread that belongs to the first block and one that belongs to the last block that might get executed much later and on a different SM. Overall synchronization can be obtained by breaking the algorithm in two kernel functions right at the point where synchronization is desired. Thus, after the execution of the first kernel the control is rendered back to the host, which upon the invocation of the subsequent kernel ensures that all threads start on equal footing. This approach is feasible since the device memory is persistent between subsequent kernel calls as long as they are made by the same host process. The strategy works albeit at a small computational cost as there is an overhead associated with each kernel call.



Specifically, the overhead of launching a kernel for execution is on average between  $90 \mu\text{s}$  (when no function arguments are present) and  $120 \mu\text{s}$  (when arguments such as pointers to device memory are passed in the kernel argument list).

Looking ahead, the next generation of GPU hardware and CUDA software will make the heterogeneous computing model, where some tasks are executed by the host and other compute intensive parts of the code are delegated to the GPU, even more attractive. Slated to be released by March 2010, the Fermi family of GPUs will have 512 SPs in one SM and up to 1 TB of fast Graphics Double Data Rate, version 5 (GDDR5) memory. Moreover, the current weak double precision performance of the GPU (about eight times slower than single precision peak performance) will be improved to clock at half the value of the single precision peak performance. Finally, on the software side, the CUDA run-time API will provide (a) support for stream computing where expensive host-device data moving operations can be overlapped with kernel execution, and (b) a mechanism to simultaneously execute on the device different kernels that are data independent. It becomes apparent that if used for the right type of applications, that is, when the execution bottleneck fits the SIMD computational model, and if used right, GPU computing can lead to impressive reductions in computational time. Combined with its affordability attribute, GPU computing will allow scientific computing to tackle large problems that in the past fell outside the realm of tractable problems. The class of granular dynamics problems is one such example, where a discrete approach to equation formulation and solution was not feasible in most cases in the past.

### 3 Large Scale Multibody Dynamics on the GPU

This section briefly introduces the theoretical background for mechanical systems made up of multiple rigid bodies whose time evolution is controlled by external forces, frictional contacts, bilateral constraints and motors.

#### 3.1 The Formulation of the Equations of Motion

The state of a mechanical system with  $n_b$  rigid bodies in three dimensional space can be represented by the generalized coordinates

$$\mathbf{q} = \left[ \mathbf{r}_1^T, \epsilon_1^T, \dots, \mathbf{r}_{n_b}^T, \epsilon_{n_b}^T \right]^T \in \mathbb{R}^{7n_b}$$

and their time derivatives

$$\dot{\mathbf{q}} = \left[ \dot{\mathbf{r}}_1^T, \dot{\epsilon}_1^T, \dots, \dot{\mathbf{r}}_{n_b}^T, \dot{\epsilon}_{n_b}^T \right]^T \in \mathbb{R}^{7n_b},$$



where  $\mathbf{r}_j$  is the absolute position of the center of mass of the  $j$ -th body and the quaternion  $\epsilon_j$  expresses its rotation. One can also introduce the generalized velocities  $\mathbf{v} = [\dot{\mathbf{r}}_1^T, \dot{\bar{\omega}}_1^T, \dots, \dot{\mathbf{r}}_{n_b}^T, \dot{\bar{\omega}}_{n_b}^T]^T \in \mathbb{R}^{6n_b}$ , directly related to  $\dot{\mathbf{q}}$  by means of the linear mapping  $\dot{\mathbf{q}} = \mathbf{L}(\mathbf{q})\mathbf{v}$  that transforms each angular velocity  $\bar{\omega}_i$  (expressed in the local coordinates of the body) into the corresponding quaternion derivative  $\dot{\epsilon}_i$  by means of the linear algebra formula  $\dot{\epsilon}_i = \frac{1}{2}G(\epsilon_j)\bar{\omega}_i$ , with

$$G(\epsilon_j) = \begin{bmatrix} +\epsilon_1 & +\epsilon_0 & -\epsilon_3 & +\epsilon_2 \\ +\epsilon_2 & +\epsilon_3 & +\epsilon_0 & -\epsilon_1 \\ +\epsilon_3 & -\epsilon_2 & +\epsilon_1 & +\epsilon_0 \end{bmatrix}.$$

Mechanical constraints, such as revolute or prismatic joints, can exist between the parts: they translate into algebraic equations that constrain the relative position of pairs of bodies. Assuming a set  $\mathcal{B}$  of constraints is present in the system, they lead to the scalar equations

$$\Psi_i(\mathbf{q}, t) = 0, \quad i \in \mathcal{B}.$$

To ensure that constraints are not violated in terms of velocities, one must also satisfy the first derivative of the constraint equations, that is

$$\nabla \Psi_i^T \mathbf{v} + \frac{\partial \Psi_i}{\partial t} = 0, \quad i \in \mathcal{B}.$$

with the Jacobian matrix  $\nabla_q \Psi_i = [\partial \Psi_i / \partial \mathbf{q}]^T$  and  $\nabla \Psi_i^T = \nabla_q \Psi_i^T \mathbf{L}(\mathbf{q})$ . Note that the term  $\partial \Psi_i / \partial t$  is null for all scleronomic constraints, but it might be nonzero for constraints that impose some trajectory or motion law, such as in the case of motors and actuators.

If contacts between rigid bodies must be taken into consideration, colliding shapes must be defined for each body. A collision detection algorithm must be used to provide a set of pairs of contact points for bodies whose shapes are near enough, so that a set  $\mathcal{A}$  of inequalities can be used to concisely express the non-penetration condition between the volumes of the shapes:

$$\Phi_i(\mathbf{q}) \geq 0, \quad i \in \mathcal{A}$$

Note that for curved convex shapes, such as spheres and ellipsoids, there is a unique pair of contact points, that is the pair of closest points on their surfaces, but in case of faceted or non-convex shapes there might be multiple pairs of contact points, whose definition is not always trivial and whose set may be discontinuous.

Given two bodies in contact  $A, B \in \{1, 2, \dots, n_b\}$  let  $\mathbf{n}_i$  be the normal at the contact pointing toward the exterior of body  $A$ , and let  $\mathbf{u}_i$  and  $\mathbf{w}_i$  be two vectors in the contact plane such that  $\mathbf{n}_i, \mathbf{u}_i, \mathbf{w}_i \in \mathbb{R}^3$  are mutually orthogonal vectors. When a contact  $i$  is active, that is, for  $\Phi_i(\mathbf{q}) = 0$ , the frictional contact force acts on the system by means of multipliers  $\hat{\gamma}_{i,n} \geq 0$ ,  $\hat{\gamma}_{i,u}$ , and  $\hat{\gamma}_{i,w}$ . Specifically, the normal component of the contact force acting on body  $B$  is  $\mathbf{F}_{i,N} = \hat{\gamma}_{i,n} \mathbf{n}_i$  and the tangential component is  $\mathbf{F}_{i,T} = \hat{\gamma}_{i,u} \mathbf{u}_i + \hat{\gamma}_{i,w} \mathbf{w}_i$  (for body  $A$  these forces have the opposite sign).

Also, according to the Coulomb friction model, in case of nonzero relative tangential speed,  $\mathbf{v}_{i,T}$ , the direction of the tangential contact force is aligned to  $\mathbf{v}_{i,T}$  and it is proportional to the normal force as  $\|\mathbf{F}_{i,T}\| = \mu_{i,d} \|\mathbf{F}_{i,N}\|$  by means of the dynamic friction coefficient  $\mu_{i,d} \in \mathbb{R}^+$ . However, in case of null tangential speed, the strength of the tangential force is limited by the inequality  $\|\mathbf{F}_{i,T}\| = \mu_{i,s} \|\mathbf{F}_{i,N}\|$  using a static friction coefficient  $\mu_{i,s} \in \mathbb{R}^+$ , and its direction is one of the infinite tangents to the surface. In our model we assume that  $\mu_{i,d}$  and  $\mu_{i,s}$  have the same value that we will write  $\mu_i$  for simplicity, so the abovementioned Coulomb model can be stated succinctly as follows:

$$\begin{aligned} \hat{\gamma}_{i,n} &\geq 0, \quad \Phi_i(\mathbf{q}) \geq 0, \quad \Phi_i(\mathbf{q})\hat{\gamma}_{i,n} = 0, \\ \mu_i \hat{\gamma}_{i,n} &\geq \sqrt{\hat{\gamma}_{i,u}^2 + \hat{\gamma}_{i,w}^2} \\ \langle \mathbf{F}_{i,T}, \mathbf{v}_{i,T} \rangle &= -\|\mathbf{F}_{i,T}\| \|\mathbf{v}_{i,T}\| \\ \|\mathbf{v}_{i,T}\| \left( \mu_i \hat{\gamma}_{i,n} - \sqrt{\hat{\gamma}_{i,u}^2 + \hat{\gamma}_{i,w}^2} \right) &= 0 \end{aligned}$$

Note that the condition  $\hat{\gamma}_{i,n} \geq 0, \Phi_i(\mathbf{q}) \geq 0, \Phi_i(\mathbf{q})\hat{\gamma}_{i,n} = 0$  can also be written as a complementarity constraint:  $\hat{\gamma}_{i,n} \geq 0 \perp \Phi_i(\mathbf{q}) \geq 0$ , see [15]. This model can also be interpreted as the Karush–Kuhn–Tucker first order conditions of the following equivalent maximum dissipation principle [6, 16]:

$$(\hat{\gamma}_{i,u}, \hat{\gamma}_{i,w}) = \underset{\sqrt{\hat{\gamma}_{i,u}^2 + \hat{\gamma}_{i,w}^2} \leq \mu_i \hat{\gamma}_{i,n}}{\operatorname{argmin}} \quad \mathbf{v}_{i,T}^T (\hat{\gamma}_{i,u} \mathbf{u}_i + \hat{\gamma}_{i,w} \mathbf{w}_i). \quad (1)$$

Finally, one should also consider the effect of external forces with the vector of generalized forces  $\mathbf{f}(t, \mathbf{q}, \mathbf{v}) \in \mathbb{R}^{6n_b}$ , that might contain gyroscopic terms, gravitational effects, forces exerted by springs or dampers, and torques applied by motors; i.e. all forces except joint reaction and frictional contact forces.

Considering the effects of both the set  $\mathcal{A}$  of frictional contacts and the set  $\mathcal{B}$  of bilateral constraints, the system cannot be reduced to either a set ordinary differential equations (ODEs) of the type  $\dot{\mathbf{v}} = \mathbf{f}(\mathbf{q}, \mathbf{v}, t)$ , or to a set of differential-algebraic equation (DAEs). This is because the inequalities and the complementarity constraints turn the system into a differential inclusion of the type  $\dot{\mathbf{v}} \in \mathcal{F}(\mathbf{q}, \mathbf{v}, t)$ , where  $\mathcal{F}(\cdot)$  is a set-valued multifunction [17]. In fact, the time evolution of the dynamical system is governed by the following differential variational inequality (DVI):

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{L}(\mathbf{q})\mathbf{v} \\ \mathbf{M}\dot{\mathbf{v}} &= \mathbf{f}(t, \mathbf{q}, \mathbf{v}) + \sum_{i \in \mathcal{B}} \hat{\gamma}_{i,b} \nabla \Psi_i \\ &\quad + \sum_{i \in \mathcal{A}} (\hat{\gamma}_{i,n} \mathbf{D}_{i,n} + \hat{\gamma}_{i,u} \mathbf{D}_{i,u} + \hat{\gamma}_{i,w} \mathbf{D}_{i,w}) \end{aligned}$$

$$\begin{aligned}
i \in \mathcal{B} : \Psi_i(\mathbf{q}, t) &= 0 \\
i \in \mathcal{A} : \hat{\gamma}_{i,n} &\geq 0 \perp \Phi_i(\mathbf{q}) \geq 0, \quad \text{and} \\
(\hat{\gamma}_{i,u}, \hat{\gamma}_{i,w}) &= \underset{\mu_i \hat{\gamma}_{i,n} \geq \sqrt{\hat{\gamma}_{i,u}^2 + \hat{\gamma}_{i,w}^2}}{\operatorname{argmin}} \mathbf{v}^T (\hat{\gamma}_{i,u} \mathbf{D}_{i,u} + \hat{\gamma}_{i,w} \mathbf{D}_{i,w})
\end{aligned} \tag{2}$$

Here, to express the contact forces in generalized coordinates, we used the tangent space generators  $D_i = [\mathbf{D}_{i,n}, \mathbf{D}_{i,u}, \mathbf{D}_{i,w}] \in \mathbb{R}^{6n_b \times 3}$  that are sparse and are defined given a pair of contacting bodies  $A$  and  $B$  as:

$$D_i^T = \begin{bmatrix} \mathbf{0} & \dots & -A_{i,p}^T & A_{i,p}^T A_A \tilde{\mathbf{s}}_{i,A} & \mathbf{0} & \dots \\ \mathbf{0} & \dots & A_{i,p}^T & -A_{i,p}^T A_B \tilde{\mathbf{s}}_{i,B} & \mathbf{0} & \dots \end{bmatrix} \tag{3}$$

Here  $A_{i,p} = [\mathbf{n}_i, \mathbf{u}_i, \mathbf{w}_i]$  is the  $\mathbb{R}^{3 \times 3}$  matrix of the local coordinates of the  $i$ -th contact, and the vectors  $\tilde{\mathbf{s}}_{i,A}$  and  $\tilde{\mathbf{s}}_{i,B}$  to represent the positions of the contact points expressed in body coordinates. The skew matrices  $\tilde{\tilde{\mathbf{s}}}_{i,A}$  and  $\tilde{\tilde{\mathbf{s}}}_{i,B}$  are defined as

$$\tilde{\tilde{\mathbf{s}}}_{i,A} = \begin{bmatrix} 0 & -s_{i,A_z} & +s_{i,A_y} \\ +s_{i,A_z} & 0 & -s_{i,A_x} \\ -s_{i,A_y} & +s_{i,A_x} & 0 \end{bmatrix}, \quad \tilde{\tilde{\mathbf{s}}}_{i,B} = \begin{bmatrix} 0 & -s_{i,B_z} & +s_{i,B_y} \\ +s_{i,B_z} & 0 & -s_{i,B_x} \\ -s_{i,B_y} & +s_{i,B_x} & 0 \end{bmatrix}$$

The DVI in (2) can be solved by time-stepping methods. The discretization requires the solution of a complementarity problem at each time step, and it has been demonstrated that it converges to the solution to the original differential inclusion for  $h \rightarrow 0$  [15, 18]. Moreover, the differential inclusion can be solved in terms of vector measures: forces can be impulsive and velocities can have discontinuities, thus supporting also the case of impacts and giving a weak solution to otherwise unsolvable situations like in the Painlevé paradox [19].

### 3.2 The Time Stepping Solver

Within the aforementioned measure differential inclusion approach, the unknowns are not the reaction forces and the accelerations  $\dot{\mathbf{v}}$  as in usual ODEs or DAEs. Instead, given a position  $\mathbf{q}^{(l)}$  and velocity  $\mathbf{v}^{(l)}$  at the time step  $t^{(l)}$ , the unknowns are the impulses  $\gamma_s$ , for  $s = n, u, w, b$  (that, for smooth constraints, can be interpreted as  $\hat{\gamma}_n = h\gamma_n, \hat{\gamma}_u = h\gamma_u, \hat{\gamma}_w = h\gamma_w, \hat{\gamma}_b = h\gamma_b$ ) and the speeds  $\mathbf{v}^{(l+1)}$  at the new time step  $t^{(l+1)} = t^{(l)} + h$ . These unknowns are obtained by solving the following optimization problem with equilibrium constraints [2]:

$$\begin{aligned}
\mathbf{M}(\mathbf{v}^{(l+1)} - \mathbf{v}^{(l)}) &= h\mathbf{f}(t^{(l)}, \mathbf{q}^{(l)}, \mathbf{v}^{(l)}) + \sum_{i \in \mathcal{B}} \gamma_{i,b} \nabla \Psi_i \\
&\quad + \sum_{i \in \mathcal{A}} (\gamma_{i,n} \mathbf{D}_{i,n} + \gamma_{i,u} \mathbf{D}_{i,u} + \gamma_{i,w} \mathbf{D}_{i,w}),
\end{aligned}$$

$$\begin{aligned}
i \in \mathcal{B} : \quad & \frac{1}{h} \Psi_i(\mathbf{q}^{(l)}, t) + \nabla \Psi_i^T \mathbf{v}^{(l+1)} + \frac{\partial \Psi_i}{\partial t} = 0 \\
i \in \mathcal{A} : \quad & 0 \leq \frac{1}{h} \Phi_i(\mathbf{q}^{(l)}) + \mathbf{D}_{i,n}^T \mathbf{v}^{(l+1)} \perp \gamma_n^i \geq 0, \\
(\gamma_{i,u}, \gamma_{i,w}) = \quad & \underset{\mu_i \gamma_{i,n} \geq \sqrt{\gamma_{i,u}^2 + \gamma_{i,w}^2}}{\operatorname{argmin}} \quad \mathbf{v}^T (\gamma_{i,u} \mathbf{D}_{i,u} + \gamma_{i,w} \mathbf{D}_{i,w}) \\
\mathbf{q}^{(l+1)} = \quad & \mathbf{q}^{(l)} + h \mathbf{L}(\mathbf{q}^{(l)}) \mathbf{v}^{(l+1)}. \tag{4}
\end{aligned}$$

The  $\frac{1}{h} \Phi_i(\mathbf{q}^{(l)})$  term is introduced to ensure contact stabilization, and its effect is discussed in [3]. Similarly, the term  $\frac{1}{h} \Psi_i(\mathbf{q}^{(l)})$  achieves stabilization for bilateral constraints.

Several numerical methods can be used to solve (4). For instance, one can approximate the Coulomb friction cones in 3D as faceted pyramids, thus leading to a LCP whose solution is possible by using off-the-shelf pivoting methods. However, these methods usually require a large computational overhead and can be used only for a limited number of variables.

Therefore, in a previous work [20] we demonstrated that the problem can be cast as a monotone optimization problem by introducing a relaxation over the complementarity constraints, replacing  $0 \leq \frac{1}{h} \Phi_i(\mathbf{q}^{(l)}) + \mathbf{D}_{i,n}^T \mathbf{v}^{(l+1)} \perp \gamma_n^i \geq 0$  with  $0 \leq \frac{1}{h} \Phi_i(\mathbf{q}^{(l)}) + \mathbf{D}_{i,n}^T \mathbf{v}^{(l+1)} - \mu_i \sqrt{(\mathbf{v}^T \mathbf{D}_{i,u})^2 + (\mathbf{v}^T \mathbf{D}_{i,w})^2} \perp \gamma_n^i \geq 0$ . The solution of the modified time stepping scheme approaches the solution of the original differential inclusion for  $h \rightarrow 0$  just as the original scheme [3]. Most importantly, the modified scheme becomes a Cone Complementarity Problem (CCP), which can be solved efficiently by an iterative numerical method that relies on projected contractive maps. Omitting for brevity some of the details discussed in [21], the algorithm makes use of the following vectors and matrices:

$$\begin{aligned}
\gamma_{i,a} &\equiv \{\gamma_{i,n}, \gamma_{i,u}, \gamma_{i,w}\}^T, \quad i \in \mathcal{A}, \\
\mathbf{b}_i &\equiv \left\{ \frac{1}{h} \Phi_i(\mathbf{q}^{(l)}), 0, 0 \right\}^T, \quad i \in \mathcal{A}, \\
b_i &\equiv \frac{1}{h} \Psi_i(\mathbf{q}^{(l)}, t) + \frac{\partial \Psi_i}{\partial t}, \quad i \in \mathcal{A} \tag{5}
\end{aligned}$$

The solution of the CCP is obtained by iterating the following expressions on  $r$  until convergence, or until  $r$  exceeds a maximum amount of iterations, starting from  $\mathbf{v}^0 = \mathbf{v}^{(l)}$ :

$$\forall i \in \mathcal{A} : \quad \gamma_{i,a}^{r+1} = \Pi_{\Upsilon_i} \left[ \gamma_{i,a}^r - \omega \eta_i \left( D_i^T \mathbf{v}^r + \mathbf{b}_i \right) \right] \tag{6}$$

$$\forall i \in \mathcal{A} : \quad \gamma_{i,b}^{r+1} = \gamma_{i,b}^r - \omega \eta_i \left( \nabla \Psi_i^T \mathbf{v}^r + \mathbf{b}_i \right) \tag{7}$$

$$\mathbf{v}^{r+1} = \mathbf{v}^r + M^{-1} \left( \sum_{z \in \mathcal{A}} D_z \gamma_{z,a}^{r+1} + \sum_{z \in \mathcal{B}} \nabla \Psi_z \gamma_{z,b}^{r+1} + h \mathbf{f}(t^{(l)}, \mathbf{q}^{(l)}, \mathbf{v}^{(l)}) \right) \quad (8)$$

Note that the superscript  $(l + 1)$  was omitted for brevity.

The iterative process uses the projector  $\Pi_{\Upsilon_i}(\cdot)$ , which is a non-expansive metric map  $\Pi_{\Upsilon_i} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  acting on the triplet of multipliers associated with the  $i$ -th contact [20]. In detail, if the multipliers fall into the friction cone

$$\Upsilon_i = \left\{ \gamma_{i,a} \in \mathbb{R}^3 : (\gamma_{i,u}^2 + \gamma_{i,w}^2)^{1/2} \leq \mu_i \gamma_{i,n} \right\}$$

they are not modified; if they are in the polar cone

$$\Upsilon_i^o = \left\{ \mathbf{x}_i \in \mathbb{R}^3 : \langle \mathbf{x}_i, \gamma_{i,a} \rangle \leq 0, \forall \gamma_{i,a} \in \Upsilon_i \right\}$$

they are set to zero; in the remaining cases they are projected orthogonally onto the surface of the friction cone. The over-relaxation factor  $\omega$  and  $\eta_i$  parameters are adjusted to control the convergence. Interested readers are referred to [21] for a proof of the convergence of this method.

For improved performance, the summation of Eq. (8) can be computed only once at the beginning of the CCP iteration, while the following updates can be performed using an incremental version that avoids adding the  $\mathbf{f}(t^{(l)}, \mathbf{q}^{(l)}, \mathbf{v}^{(l)})$  term all the time; in case there is no initial guess for the multipliers and  $\gamma_{i,b}^0 = 0, \gamma_{i,a}^0 = 0$ , Eq. (8) turns into:

$$\mathbf{v}^0 = \mathbf{v}^{(l)} + M^{-1} h \mathbf{f}(t^{(l)}, \mathbf{q}^{(l)}, \mathbf{v}^{(l)}) \quad (9)$$

$$\mathbf{v}^{r+1} = \mathbf{v}^r + \sum \Delta \mathbf{v}_i \quad (10)$$

where

$$i \in \mathcal{A} : \quad \Delta \mathbf{v}_i = \sum_{i \in \mathcal{A}} M^{-1} D_i \Delta \gamma_{i,a}^{r+1}$$

$$i \in \mathcal{B} : \quad \Delta \mathbf{v}_i = \sum_{i \in \mathcal{B}} M^{-1} \nabla \Psi_i \Delta \gamma_{i,b}^{r+1}$$

In the case that only bilateral constraints are used, this method behaves like the typical fixed-point Jacobi iteration for the solution of linear problems. If one interleaves the update (8) after each time that a single  $i$ -th multiplier is computed in (6) or (7), the resulting scheme behaves like a Gauss–Seidel method. This variant can benefit from the use of Eq. (10) instead of Eq. (8) because it can increment only the  $\Delta \mathbf{v}_i$  term corresponding to the constraint that has been just computed. Also, this immediate update of the speed vector provides better properties of convergence (especially in case of redundant constraints) but it does not fit well in a parallel computing environment because of its inherently sequential nature.

### 3.3 The GPU Formulation of the CCP Solver

Since the CCP iteration is a computational bottleneck of the numerical solution proposed, a great benefit will follow from an implementation that can take advantage of the parallel computing resources available on GPU boards.

In the proposed approach, the data structures on the GPU are implemented as large arrays (*buffers*) to match the execution model associated with NVIDIA's CUDA. Specifically, threads are grouped in rectangular thread blocks, and thread blocks are arranged in rectangular grids. Four main buffers are used: the contacts buffer, the constraints buffer, the reduction buffer, and the bodies buffer. Since repeated transfers of large data structures can adversely impact the performance of the entire algorithm, an attempt was made to organize the data structures in a way that minimized the number of fetch and store operations and maximized the arithmetic intensity of the kernel code. This ensures that the latency of the global memory can be hidden by the hardware multithread scheduler if the GPU code interleaves the memory access with enough arithmetic instructions.

Figure 5 shows the data structure for contacts, which contains two pointers  $B_A$  and  $B_B$  to the two touching bodies. There is no need to store the entire  $D_i$  matrix for the  $i$ -th contact because it has zero entries everywhere except for the two  $12 \times 3$  blocks corresponding to the coordinates of the two bodies in contact. In detail, we store only the following  $3 \times 3$  matrices:

$$\begin{aligned} D_{i,v_A}^T &= -A_{i,p}^T, & D_{i,\omega_A}^T &= A_{i,p}^T A_A \tilde{s}_{i,A} \\ D_{i,v_B}^T &= A_{i,p}^T, & D_{i,\omega_B}^T &= -A_{i,p}^T A_B \tilde{s}_{i,B} \end{aligned}$$

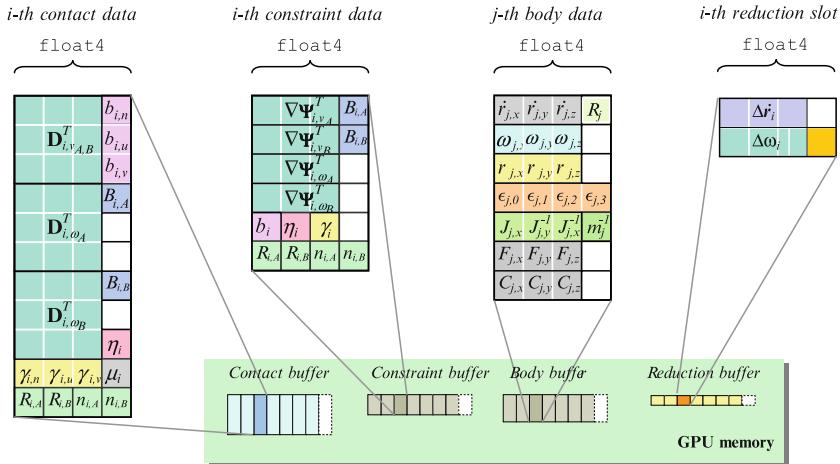


Fig. 5 Data structures in GPU global memory

Once the velocities of the two bodies  $\dot{\mathbf{r}}_{A_i}, \bar{\omega}_{A_i}, \dot{\mathbf{r}}_{B_i}$  and  $\bar{\omega}_{B_i}$  have been fetched, the product  $D_i^T \mathbf{v}^r$  in Eq. (6) can be performed as

$$D_i^T \mathbf{v}^r = D_{i,v_A}^T \dot{\mathbf{r}}_{A_i} + D_{i,\omega_A}^T \bar{\omega}_{A_i} + D_{i,v_B}^T \dot{\mathbf{r}}_{B_i} + D_{i,\omega_B}^T \bar{\omega}_{B_i} \quad (11)$$

Since  $D_{i,v_A}^T = -D_{i,v_B}^T$ , there is no need to store both matrices, so in each contact data structure only a matrix  $D_{i,v_{AB}}^T$  is stored, which is then used with opposite signs for each of the two bodies.

Also, the velocity update vector  $\Delta \mathbf{v}_i$ , needed for the sum in Eq. (10) is sparse: it can be decomposed in small  $3 \times 1$  vectors. Specifically, given the masses and the inertia tensors of the two bodies  $m_{A_i}, m_{B_i}, J_{A_i}$  and  $J_{B_i}$ , the term  $\Delta \mathbf{v}_i$  will be computed and stored in four parts as follows:

$$\begin{aligned} \Delta \dot{\mathbf{r}}_{A_i} &= m_{A_i}^{-1} D_{i,v_A} \Delta \gamma_{i,a}^{r+1}, & \Delta \bar{\omega}_{A_i} &= J_{A_i}^{-1} D_{i,\omega_A} \Delta \gamma_{i,a}^{r+1} \\ \Delta \dot{\mathbf{r}}_{B_i} &= m_{B_i}^{-1} D_{i,v_B} \Delta \gamma_{i,a}^{r+1}, & \Delta \bar{\omega}_{B_i} &= J_{B_i}^{-1} D_{i,\omega_B} \Delta \gamma_{i,a}^{r+1} \end{aligned} \quad (12)$$

Note that those four parts of the  $\Delta \mathbf{v}_i$  terms are not stored in the  $i$ -th contact or data structures of the two referenced bodies (because multiple contacts may refer the same body, hence they would overwrite the same memory position). These velocity updates are instead stored in the reduction buffer, which will be used to efficiently perform the summation in Eq. (10). This will be discussed shortly.

The constraints buffer, shown in Fig. 5, is based on a similar concept. Jacobians  $\nabla \Psi_i$  of all scalar constraints are stored in a sparse format, each corresponding to four rows  $\nabla \Psi_{i,v_A}, \nabla \Psi_{i,\omega_A}, \nabla \Psi_{i,v_B}, \nabla \Psi_{i,\omega_B}$ . Therefore the product  $\nabla \Psi_i^T \mathbf{v}^r$  in Eq. (7) can be performed as the scalar value:

$$\nabla \Psi_i^T \mathbf{v}^r = \nabla \Psi_{i,v_A}^T \dot{\mathbf{r}}_{A_i} + \nabla \Psi_{i,\omega_A}^T \bar{\omega}_{A_i} + \nabla \Psi_{i,v_B}^T \dot{\mathbf{r}}_{B_i} + \nabla \Psi_{i,\omega_B}^T \bar{\omega}_{B_i} \quad (13)$$

Also, the four parts of the sparse vector  $\Delta \mathbf{v}_i$  can be computed and stored as

$$\begin{aligned} \Delta \dot{\mathbf{r}}_{A_i} &= m_{A_i}^{-1} \nabla \Psi_{i,v_A} \Delta \gamma_{i,b}^{r+1}, & \Delta \bar{\omega}_{A_i} &= J_{A_i}^{-1} \nabla \Psi_{i,\omega_A} \Delta \gamma_{i,b}^{r+1} \\ \Delta \dot{\mathbf{r}}_{B_i} &= m_{B_i}^{-1} \nabla \Psi_{i,v_B} \Delta \gamma_{i,b}^{r+1}, & \Delta \bar{\omega}_{B_i} &= J_{B_i}^{-1} \nabla \Psi_{i,\omega_B} \Delta \gamma_{i,b}^{r+1} \end{aligned} \quad (14)$$

Figure 5 shows that each body is represented by a data structure containing the state (velocity and position), the mass moments of inertia and mass values, and the external applied force  $\mathbf{F}_j$  and torque  $\mathbf{C}_j$ . Those data are needed to compute the CCP iteration and solve for unknowns.

When it comes to the implementation of the CCP solver on the GPU, using kernels that operate on the abovementioned data buffers, the task is not trivial because the iteration cannot be performed with a single kernel. In fact, considering the iteration over Eqs. (6), (7), and (10), one can see that Eqs. (6) and (7) fit into parallel kernels that operate, respectively, one thread per contact and one thread per bilateral constraint. Moreover, the summation in Eq. (10) cannot be easily parallelized

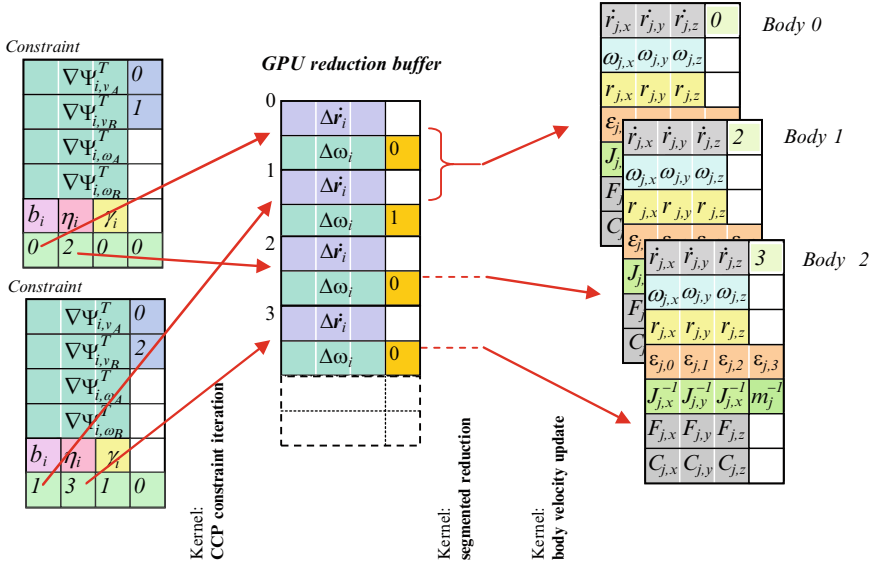


Fig. 6 Example of reduction buffer for summing up body velocities

in the same way because it may happen that two or more contacts need to add their velocity updates  $\Delta \mathbf{v}_i$  to the same rigid body: this would cause a race condition where multiple threads might need to update the same memory value, something that can cause errors or indefinite/nondeterministic behaviors on the GPU hardware. Therefore, in order to parallelize Eq. (10), a parallel segmented scan algorithm [22] was adopted that operates on an intermediate reduction buffer (see Fig. 6); this method sums the values in the buffer using a binary-tree approach that keeps the computational load well balanced among the many processors. In the example of Fig. 6, the first constraint refers to bodies 0 and 1, the second to bodies 0 and 2; multiple updates to body 0 are then accumulated with parallel a segmented reduction.

Note that several other auxiliary kernels that have minimal impact on the computation time are used to prepare pre-process data before the CCP starts, for example to compute Eq. (9). Also, to speed up the computation, matrices  $D_{i,v,A}^T$ ,  $D_{i,\omega,A}^T$  and  $D_{i,\omega,B}^T$  are not provided by the host; instead they are computed on the GPU using the data coming from the collision detection code, that is,  $\bar{\mathbf{s}}_{i,A}$ ,  $\bar{\mathbf{s}}_{i,B}$  and  $\mathbf{n}_i$ .

The following pseudocode shows the sequence of the main computational stages at each time step, which for the most part are executed as parallel kernels on the GPU (Table 2).

Stages 1 and 10 can be avoided if one manages to keep all the data on the GPU, by letting the collision detection engine communicate with the CCP solver directly. Even if those memory transfers are executed only at the beginning and at the end of the CCP solution process, their impact on the overall simulation time might be significant.



**Table 2** Pseudocode for the CCP solver

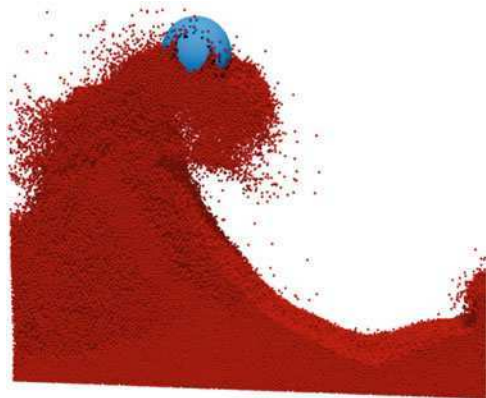
Stage	Context	Operations/kernels
1	HOST Serial	Copy memory CPU→ GPU Copy contact and body data structures from host memory to GPU buffers Copy constraint data (residuals $b_i$ and Jacobians) into the constraint buffer
2	GPU Parallel on bodies	Force kernel For each body, compute forces $\mathbf{f}^{(l)}$ , $\mathbf{q}^{(l)}$ , $\mathbf{v}^{(l)}$ , if any. Store these forces and torques into $\mathbf{F}_j$ and $\mathbf{C}_j$
3	GPU Parallel on contacts	Contact preprocessing kernel For each contact, given contact normal and position, compute in place the matrices $D_{i,vA}^T$ , $D_{i,\omega A}^T$ and $D_{i,\omega B}^T$ , then compute $\eta_i$ and the contact residual $\mathbf{b}_i = \{\frac{1}{h}\Phi_i(\mathbf{q}), 0, 0\}^T$
4	GPU Parallel on bodies	CCP force kernel For each body $j$ , initialize body velocities: $\dot{\mathbf{r}}_j^{(l+1)} = h m_j^{-1} \mathbf{F}_j$ and $\dot{\omega}_j^{(l+1)} = h J_j^{-1} \mathbf{C}_j$
5	GPU Parallel on contacts	CCP contact iteration kernel For each contact $i$ , do $\gamma_{i,a}^{r+1} = \Pi_{\gamma_i}[\gamma_{i,a}^r - \omega \eta_i (D_i^T \mathbf{v}^r + \mathbf{b}_i)]$  Note that $D_i^T \mathbf{v}^r$ is evaluated with sparse data, using Eq. (11) Store $\Delta \gamma_{i,a}^{r+1} = \gamma_{i,a}^{r+1} - \gamma_{i,a}^r$ in contact buffer. Use Eq. (12) to compute sparse updates $\Delta \dot{\mathbf{r}}$ and $\Delta \dot{\omega}$ to the velocities of the two connected bodies $A$ and $B$ , and store them in the $R_{i,A}$ and $R_{i,B}$ slots of the reduction buffer
6	GPU Parallel on constraints	CCP constraint iteration kernel For each constraint $i$ , do $\gamma_{i,b}^{r+1} = \gamma_{i,b}^r - \omega \eta_i (\nabla \Psi_i^T \mathbf{v}^r + b_i)$ .  Note that $\nabla \Psi_i^T \mathbf{v}^r$ is evaluated with sparse data, using Eq. (11) Store $\Delta \gamma_{i,b}^{r+1} = \gamma_{i,b}^{r+1} - \gamma_{i,b}^r$ in contact buffer. Use Eq.(14) to compute sparse updates $\Delta \dot{\mathbf{r}}$ and $\Delta \dot{\omega}$ to the velocities of the two connected bodies $A$ and $B$ , and store them in the $R_{i,A}$ and $R_{i,B}$ slots of the reduction buffer
7	GPU Parallel on reduction slots	Segmented reduction kernel Sum all the $\Delta \dot{\mathbf{r}}$ and $\Delta \dot{\omega}$ terms belonging to the same body, in the reduction buffer. This may require a sequence of short kernels
8	GPU Parallel on bodies	Body velocity updates kernel For each $j$ body, add the cumulative velocity updates which can be fetched from the reduction buffer, using the index $R_j$
9	HOST Serial	Check convergence and repeat from stage 5 if convergence tolerance is not reached
10	HOST Serial	Copy memory GPU→ CPU Copy contact multipliers from GPU buffers to host memory, if interested in reaction forces Copy constraints multipliers from GPU buffers to host memory, if interested in reaction forces Copy rigid body velocities from GPU buffers to host memory

## 4 Numerical Experiments

The largest simulation run to date using the CCP-based GPU solver discussed herein contained approximately 1.1 million bodies that interacted through frictional contact as illustrated in Fig. 7. This problem has a large number of small spheres made up of a material with high density. There is one large ball of low density mixed up with the rest of the spheres. The collection of balls is inside a three dimensional rectangular box that experiences a left-to-right harmonic motion. Because the large ball has lower density, it will eventually “float” on the spheres of high density as illustrated in the figure. This test, along with other simulations focused on tracked vehicle mobility on granular terrain are discussed in detail in [23, 24]. An animation is available at [25].

In what follows the emphasis is on a comparison between the GPU-based solution and a sequential approach used to solve a benchmark problem; i.e., the flow of a pebble bed nuclear reactor (Fig. 8). The fuel is encased in tennis-ball-size graphite spheres, each filled with nuclear fuel, specifically, coated  $\text{UO}_2$ , with sub-millimeter diameter [26]. The approximately 400,000 pebbles are continuously recirculated or refreshed at a rate of about 2/min [27]. They are densely packed, at volume fractions approaching 0.6, and thus constitute a dense granular flow [28]. The center pebbles, represented with a different color, are moderator pebbles with comparable weight to the fuel pebbles, even if they do not contain particles of coated  $\text{UO}_2$ . The reactor is cooled with a fast helium flow blown top-down that has negligible drag effects on the spheres when compared to gravitational forces [28]. Predicting the dynamics of the fuel pebbles in the pebble-bed reactor is important for its safety and gauging its performance [29].

To better understand the potential of parallel computing when employed to solve the problem at hand, both the sequential and parallel implementations draw on the same solution procedure detailed in Section 3. The only difference is that in one case the collision detection and the solution of the cone-complementarity problem are carried out sequentially, on the CPU, while for the parallel implementation these



**Fig. 7** Largest problem simulated to date, the system has about 1.1 million bodies that are shaken in a moving box

**Fig. 8** Pebble bed nuclear reactor simulation

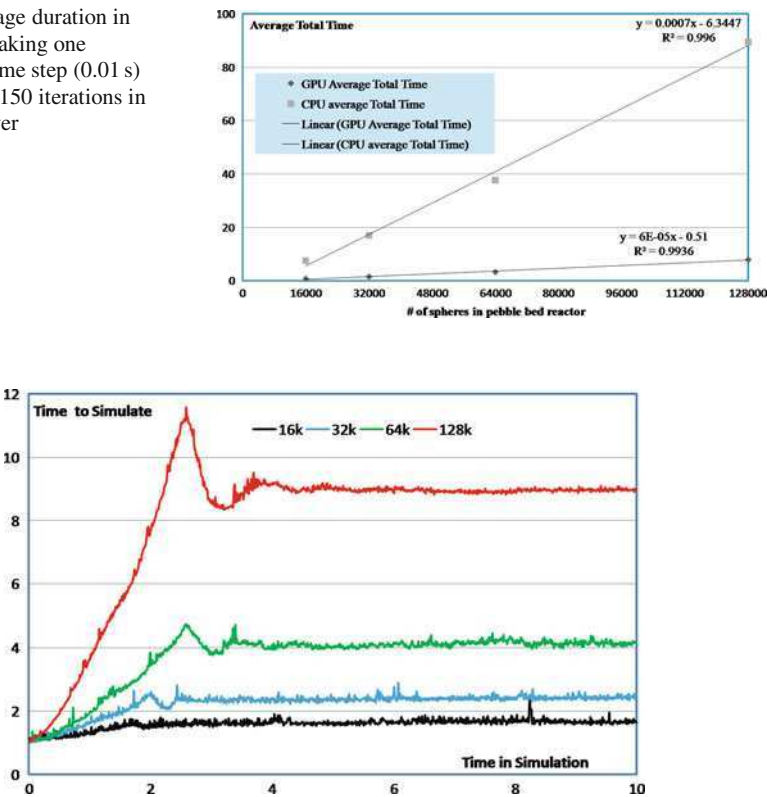


two stages, along with several other less computationally intensive steps of the solution methodology, are executed on the GPU. The benchmark problem was run for a set of 16, 32, 64, and 128 thousand particles. The sequential simulation was run on a single threaded Quad Core Intel Xeon E5430 2.66 GHz computer. For the parallel version, the collision detection was implemented on a NVIDIA 8800 GT card, while the cone complementarity problem was solved on a Tesla C870. The integration time step considered for this problem was 0.01 s. A number of 150 iterations was considered in the solution of the CCP problem.

The dynamics of the pebble flow is as follows. First, the silo is closed and the balls are dropped from the top until the desired number of spheres is reached. The silo is subsequently opened, at which time the pebble flow commences. Shortly thereafter the flow reaches a steady state. At this time, the amount of time it takes to advance the simulation by one time step is measured. An average of this value obtained over several simulations is reported in Fig. 9. This process is carried out for both the GPU and CPU implementations for each of the four scenarios (16,000–128,000 bodies). The plot reveals that (a) both the CPU and GPU implementations scale linearly with the number of bodies in the problem, and (b) the slope of the plot associated with the GPU implementation is smaller than that associated with the CPU solver. In fact, for 128,000 particles, the GPU solver is about 10 times faster than the CPU solver. As the interest is in multi-million body problems, this slope difference will result in significant reduction in simulation times.

The plot in Fig. 10 provides the history for the amount of time it took the GPU solver to perform one integration step. In the beginning, when the balls are filling up the silo, there are few contacts and one integration time step is cheap. As the number of spheres in contact increases due to piling up of the bodies at the bottom of the silo, the time it takes to complete one time step increases. This is due to the gradual increase in the dimension of the CCP problem that needs to be solved. An artifact of the fact that only 150 iterations were considered in the CCP problem is the spurious increase (the peak) that is more pronounced for the 128,000 body case.

**Fig. 9** Average duration in seconds for taking one integration time step (0.01 s) when taking 150 iterations in the CCP solver



**Fig. 10** For the GPU solver, the plot shows how much time it took the solver to advance the simulation by one time step. On the horizontal axis is shown the simulation time. After approximately 3 seconds, when the flow reaches a steady state, each time step takes about the same amount of time, that is, approximately 9 s. These times were used in Fig. 9 to generate the lower curve (one with smaller slope)

This was intentionally kept in order to demonstrate what happens if the CCP problem is not solved accurately. Specifically, if the number of iterations is not enough to lead to the convergence of the CCP solver, the amount of penetration between bodies will increase leading to a larger number of contacts and therefore a larger CCP problem. However, as the bottom of the silo opens up, the bodies start falling and this regime is less challenging for the solver since the number of contacts suddenly decreases until reaching a steady state shortly after 3 s. At that point the amount of time required to advance the simulation by one time step stabilizes. Note that the average of this value over several simulations was used in Fig. 9 to generate the plot associated with the GPU-based solution.

## 5 Conclusions and Directions of Future Work

Two observations justify today a fresh look at the solution of frictional contact problems in large scale multibody dynamics simulation. First, existing graphics cards provide tremendous flop rate at very low costs. Second, there is a wide spectrum of real-life applications that lead to large frictional contact dominated multibody dynamics problems that couldn't be solved in the past. The contribution of this paper is in presenting an approach for parallel computational multibody dynamics that can be used to tackle many of these applications. The frictional contact problem was formulated in a form that is suitable to be numerically solved in parallel and could take advantage of commodity parallel computing support available on the GPU. The collision detection and solution of the cone complementarity problem turned out to be the main computational bottlenecks of the simulation. Both these stages of the solution have been parallelized thus enabling the implementation of an approach that can tackle problems with more than one million bodies.

Ongoing projects are aimed at: (a) increasing the size of the problem that can be solved by the proposed approach, (b) improving the speed of convergence of the CCP solver, (c) establishing the hardware infrastructure that can support the simulation of multibody dynamics problems with tens to hundreds of millions of bodies, and (d) performing an experimental validation of the simulation approach proposed. In terms of (a), current numerical experiments revealed that the 4 GB memory on the Tesla C1060 cards is exhausted for simulations that exceed 1.1 million bodies. A domain decomposition approach is anticipated to further increase this number by distributing a large problem to multiple GPUs using a spatial domain decomposition idea. The net outcome of this approach will be a pooling together of the memory resources of multiple cards. In terms of (b), it is anticipated that algebraic multi-grid methods will enable a reduction of the number of iterations required for convergence. Unless this issue gets addressed, problems with tens of millions of bodies might require prohibitively long convergence times that render the approach infeasible. In terms of (c), a 21 SP Teraflop cluster is currently assembled at the University of Wisconsin to support the domain decomposition approach described. The cluster will have one head node and six compute nodes, each of the latter with four Tesla C1060 NVIDIA GPUs. Finally, experimental validation is currently carried out both at macroscale, using 5 mm plastic particles, and microscale, using 100 and 500  $\mu\text{m}$  glass spheres, respectively. In both cases, the experiments measure flow rates in silo replicas and a small hopper to validate the correctness of the simulation results. In addition to these four initiatives, there is a multitude of small projects that remain to be completed, the most important of which being the integration of the collision detection and CCP data structures. Currently, data is moved back and forth between the device and host right after the collision detection and before performing the CCP. This adds a significant overhead that once eliminated is anticipated to further improve the performance of the GPU solver.

## References

1. Pfeiffer F, Glocker C (1996) *Multibody dynamics with unilateral contacts*. Wiley, Singapore
2. Anitescu M (2006) Optimization-based simulation of nonsmooth dynamics. *Math Programming* 105(1): 113–143
3. Anitescu M, Hart GD (2004) A constraint-stabilized time-stepping approach for rigid multi-body dynamics with joints, contact and friction. *Int J Numer Methods Eng* 60(14): 2335–2371
4. Anitescu M, Potra FA, Stewart DE (1999) Time-stepping for three-dimensional rigid body dynamics. *Comput Methods Appl Mech Eng* 177(3–4): 183–197
5. Lotstedt P (1982) Mechanical systems of rigid bodies subject to unilateral constraints. *SIAM J Appl Math* 42(2): 281–296
6. Marques M (1993) *Differential inclusions in nonsmooth mechanical problems: shocks and dry friction*. Birkhäuser, Boston, MA
7. Moreau JJ (1983) Standard inelastic shocks and the dynamics of unilateral constraints: CISM Courses and Lectures. In: Piero GD, Macieri F (eds) *Unilateral problems in structural analysis*. Wiley, New York, p 173–221
8. Pang JS, Kumar V, Song P (2005) Convergence of time-stepping method for initial and boundary-value frictional compliant contact problems. *SIAM J Numer Anal* 43: 2200
9. Pang, JS, Trinkle JC (1996) Complementarity formulations and existence of solutions of dynamic multi-rigid-body contact problems with Coulomb friction. *Math Programming* 73(2): 199–226.
10. Song P, Kraus P, Kumar V, Dupont P (2001) Analysis of rigid-body dynamic models for simulation of systems with frictional contacts. *J Appl Mech* 68(1): 118–128
11. Glocker C, Pfeiffer F (1995) Multiple impacts with friction in rigid multibody systems. *Non-linear Dyn* 7(4): 471–497
12. Manferdelli JL (2007) The many-core inflection point for mass market computer systems. *CTWatch Quart* 3(1)
13. Negrut D (2008) High performance computing for engineering applications, Course Notes ME964 (September 9 Lecture): <http://sbel.wisc.edu/Courses/ME964/2008/index.htm>, University of Wisconsin
14. NVIDIA (2009) Compute unified device architecture programming guide 2.3: [http://developer.download.nvidia.com/compute/cuda/2\\_3/toolkit/docs/NVIDIA\\_CUDA\\_ProgrammingGuide\\_2.3.pdf](http://developer.download.nvidia.com/compute/cuda/2_3/toolkit/docs/NVIDIA_CUDA_ProgrammingGuide_2.3.pdf).
15. Stewart DE, Trinkle JC (1996) An implicit time-stepping scheme for rigid-body dynamics with inelastic collisions and Coulomb friction. *Int J Numer Methods Eng* 39: 2673–2691
16. Moreau J (1988) Unilateral contact and dry friction in finite freedom dynamics. *Nonsmooth Mech Appl*, 302: 1–82
17. Pfeiffer F, Foerg M, Ulbrich H (2006) Numerical aspects of non-smooth multibody dynamics. *Comput Methods Appl Mech Eng* 195(50–51): 6891–6908
18. Stewart DE (2000) Rigid-body dynamics with friction and impact. *SIAM Rev* 42(1): 3–39
19. Stewart DE (1998) Convergence of a time stepping scheme for rigid body dynamics and resolution of Painlevé’s problem. *Arch Ration Mech Anal* 145(3): 215–260
20. Tasora A, Anitescu M (2008) A fast NCP solver for large rigid-body problems with contacts, friction, and joints. *Multibody dynamics: computational methods and applications*. Springer, Berlin, p. 45
21. Anitescu M, Tasora A (2010) An iterative approach for cone complementarity problems for nonsmooth dynamics. *Comput Optim Appl* 47(2): 207–235
22. Harris M, Shubhabrata S, Owens JD (2008) Parallel Prefix Sum (Scan) with CUDA. In: Nguyen H (ed) *GPU Gems 3*, Addison-Wesley, New York, p. 851–876
23. Heyn T, Mazhar H, Negrut D (2009) On the simulation of tracked vehicles operating on granular terrain: a parallel multibody dynamics approach (to be submitted). *Multibody system dynamics*
24. Heyn T (2009) Simulation of tracked vehicles on granular terrain leveraging GPU computing. M.S. Thesis, in Mechanical Engineering, University of Wisconsin-Madison, Madison

25. Mazhar H (2009) Million body simulation. <http://sbel.wisc.edu/Animations/index.htm>
26. Gougar H, Ougouag A, Terry W (2004) Advanced core design and fuel management for pebble-bed reactors. Idaho National Engineering and Environmental Laboratory, INEEL/EXT-04-02245
27. Kadak A, Bazant M (2004) Pebble flow experiments for pebble bed reactors, 2nd International Topical Meeting on High Temperature Reactor Technology, Beijing, China, 22–24 Sept 2004
28. Tasora A, Anitescu M (2010) A convex complementarity approach for simulating large granular flows. *J Comput Nonlinear Dynam* 5(3): 031004
29. Ougouag A, Ortensi J, Hiruta H (2009) Analysis of an earthquake-initiated-transient in a PBR. Tech. Rep. INL/CON-08-14876, Idaho National Laboratory (INL)

# Investigation of Gears Using an Elastic Multibody Model with Contact

Pascal Ziegler and Peter Eberhard

**Abstract** The classical approach to simulate contacts between gears is to use rigid body models coupled with a parallel spring damper combination. However, these models had been developed for properly meshing gears with smooth contacts and cannot cover wave propagation caused by hard contacts or impacts. Moreover, as they are based on the assumption of rigidness, often light weight designs, resulting in very compliant gear bodies, cannot be considered appropriately. To evaluate how appropriate these rigid body models are to simulate impact forces, a very detailed finite element model is used to simulate several impacts and the results are compared to simulations with a rigid body model. The results reveal that for compliant gear bodies, there exist dynamic effects that considerably affect contact forces and motion and that these effects cannot be covered by rigid body models at all. Hence, a flexible model is imperative to precisely simulate impact forces. To reduce integration time, we present a modally reduced elastic multibody model including contact that allows very precise simulations in reasonable time. For the contact calculations a node-to-segment penalty formulation is introduced and is integrated using central differences. Even though the elastic model is a reduced model, it is still of huge size, as any node on any flank is a potential contact node. Also, the transformation data between modal and nodal coordinates must be accessible during integration. To reduce the required amount of memory a coarse collision detection is introduced that allows to dynamically reload only the transformation required in the current integration step. This approach allows very precise simulations of contacts between gears with integration times about 400 times faster than for associated finite element simulations. At the same time the model is robust and fast enough to allow the simulation of many contacts and many revolutions. To validate this approach basic experimental investigations with simple impact bodies have been carried out. The results from these experiments and related simulations agree very well.

---

P. Ziegler (✉) and P. Eberhard

Institute of Engineering and Computational Mechanics, University of Stuttgart,  
Pfaffenwaldring 9, 70569 Stuttgart, Germany

e-mail: [ziegler@itm.uni-stuttgart.de](mailto:ziegler@itm.uni-stuttgart.de); [eberhard@itm.uni-stuttgart.de](mailto:eberhard@itm.uni-stuttgart.de)



## 1 Introduction

The necessity to simulate systems with gears is mainly driven by two objectives. First, to simulate the dynamical interaction of gears with other system components or to determine joint loads and, second, to calculate stresses in gears. The first objective is often approached by using rigid body models, like [5], since they are simple to set up and fast due to their small number of degrees of freedom.

The investigation of stresses requires an elastic model and very often finite element models are used in this context. To obtain meaningful stresses one has to specify realistic load cases and very often, these load cases are calculated using the aforementioned rigid body models in global overall dynamic systems.

However, if the gears are very compliable the separation of both calculations may not be valid anymore since the elasticity of the gear may have significant influence on the motion and thus on the contact forces, too. Hence, for compliable gears an elastic model is necessary right from the start to get physically correct and usable results.

## 2 Classical Models

To quantify the influence of the elasticity of gears on contact forces, a gear pair from a commercially used real gear train is investigated for several contacts. The gear pair consists of a crank shaft gear and an idler gear, see Fig. 1. This particular gear pair is mainly chosen due to the very compliable idler gear. The impact-like contacts typical for gear trains are likely to excite higher eigenfrequencies than those that are usually considered. Each gear is supported by a perfect rotational joint and both gears are assembled to initially touch in their pitch points, while the idler gear is given an initial rotational velocity of  $\omega_{i0}$ .

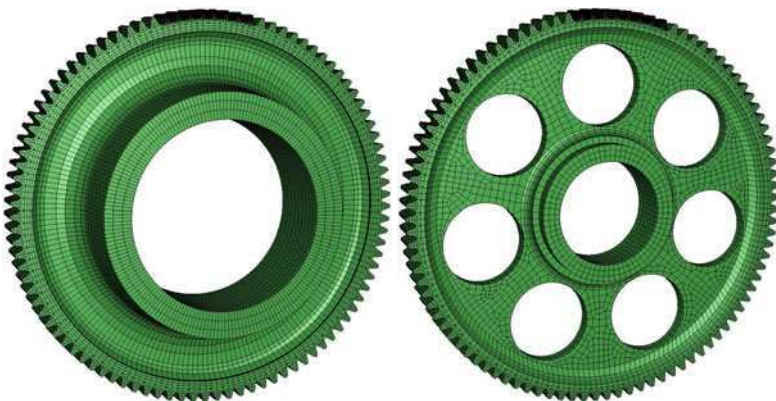


Fig. 1 Finite element models of crank shaft gear and idler gear

## 2.1 Finite Element Model

Figure 1 shows the finite element meshes of the crankshaft gear (*left*) and the idler gear (*right*). Both gears are meshed using linear hexahedral elements only. To represent the flank profiles precisely, the profiles are modeled by copying the actual production process during pre-processing, see [8]. In the contact areas an element edge length of about 0.2 mm is used to guarantee a good resolution of contact stresses and contact forces. In the gear body, however, the edge length is about 8 mm to be able to account for wave phenomena up to a frequency of 80 kHz, see [1]. This results in a total number of 145,000 elements.

The rotational joints are modeled by kinematically constraining all bore nodes to a reference node in the gear center that has only one rotational degree of freedom. For the material a fully elastic linear material model with Young's modulus  $E = 210,000 \text{ Nmm}^2$ , density  $\rho = 7,850 \text{ kg/m}^3$  and Poisson ratio  $\mu = 0.3$  is used. Damping is neglected. The finite element model is set up in Abaqus and integrated using the explicit central-difference based algorithm Abaqus/Explicit. The contact is modeled using the kinematic contact algorithm provided by Abaqus/Explicit.

## 2.2 Rigid Body Model

The simple multibody model for the gear pair consists of two rigid bodies, each having one rotational degree of freedom. The inertia properties used here are derived from the undeformed configuration of the finite element model and can, therefore, be considered to be very precise. Both bodies are coupled with a typical spring-damper combination as included in many commercial programs such that the associated forces act on the base radii  $r_{b1}$  and  $r_{b2}$ , respectively, see Fig. 2.

The backlash element in series to the spring element accounts for the unilateral contact. Damping is fully neglected, hence  $d = 0$  and the equations of motion follow as

$$I_i \ddot{\varphi}_i + r_{bi} c (r_{b1} \varphi_1 + r_{b2} \varphi_2 - e(t)) = 0, \quad i = 1, 2 \quad (1)$$

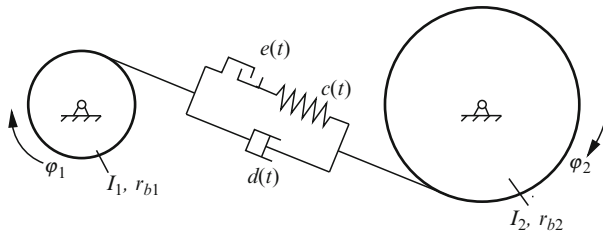


Fig. 2 Rigid body model of a gear pair

where the backlash function  $e(t)$  reads

$$e(t) = \begin{cases} 0 & \text{for } (r_{b1}\varphi_1 + r_{b2}\varphi_2) \geq 0, \\ x & \text{for } (r_{b1}\varphi_1 + r_{b2}\varphi_2) < 0. \end{cases} \quad (2)$$

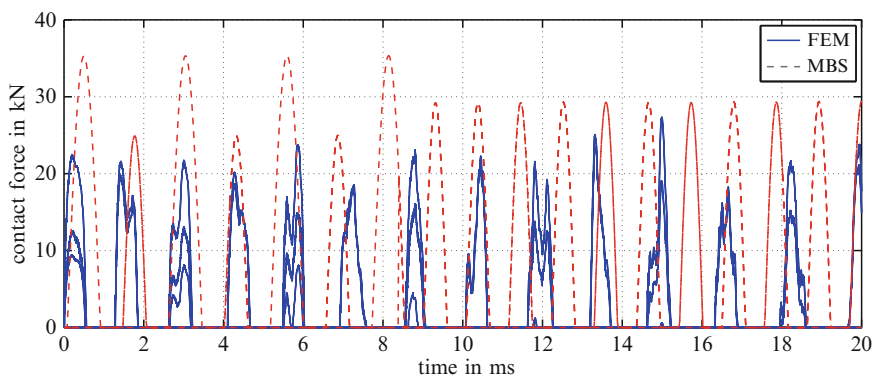
A good choice for the stiffness  $c$  is very important for good results, since this parameter describes all elasticities of the gear. According to the industrial standard DIN 3990-1, here, the so called single stiffness  $c'$ , see [6], is used. Even though there exist industrial standards like [3] or [4] to calculate the single stiffness, the values can vary significantly, see [14]. Therefore, the single stiffness is determined from a static finite element analysis using the finite element model described above and follows as

$$c = c' = 5.68 \frac{\text{N}}{\mu\text{m mm}}, \quad (3)$$

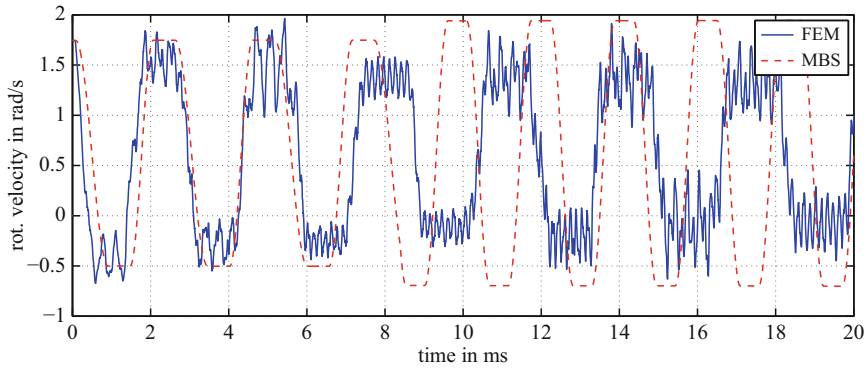
see [14] for details. The rigid body model is set up using a commercial multibody software and the provided gear contact force element is used and the parameters for the force routine are chosen such that they represent the model above. The model is integrated using the implicit Radau 5 scheme.

### 2.3 Comparison

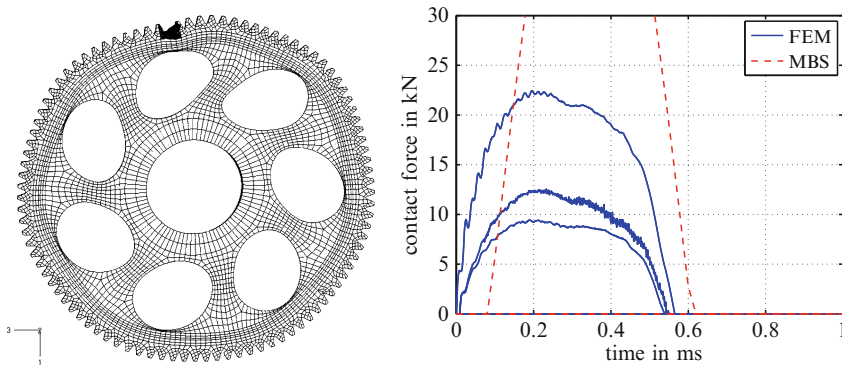
Figure 3 shows the simulated contact forces for an initial rotational velocity of  $\omega_{i0} = 100^\circ/\text{s}$ . Apparently, the contact forces calculated with the rigid body model are much higher than those calculated with the finite element model. In addition, the contact forces from the finite element model shows strong vibrations during contact. Considering the finite element model to be very precise, hence a reference solution, the rigid body model simply gives completely wrong results.



**Fig. 3** Contact forces for the rigid body model and the finite element model



**Fig. 4** Rotational velocities for the rigid body model and the finite element model



**Fig. 5** Several elastic effects: Polygonalization and torsional oscillation (*left*) and multiple pairs of cogs in contact

The rotational velocities, shown in Fig. 4, show considerable differences, too. Particularly at times between impacts, the rotational velocities from the finite element simulations show a very noticeable vibration that can not be observed from the rigid body model.

The differences in contact forces and rotational velocity are direct results of the elasticity of the gear body that cannot be sufficiently considered in a rigid body model even with refined force laws. During impact, the outer gear rim deforms under the contact force while some of the energy is stored in the deformation. This deformation can be seen from Fig. 5 (*left*).

One result of this deformation is that the contact force decreases shortly to increase again when the deformation energy is restored. A second result is that due to deformation the pitch changes to such an extent that for the investigated contact situation three pairs of cogs come into contact, while for a rigid body, theoretically only one pair is possible, see Fig. 5 (*right*). That explains that the contact

forces of the rigid body model are consistently too high. Another consequence of the deformation, shown in Fig. 5 (*left*) is the rotational vibration between contacts. The mass of the idler gear is mainly concentrated in the hub rim and the outer rim while the inner gear body is very compliant. Therefore, the impact excites an out of phase torsional vibration of the outer rim against the hub rim that can particularly be seen between contacts.

The results clearly show that for compliant gears an elastic model is absolutely necessary for precise simulations of contact forces, but regarding the integration time of about 8 days on a Pentium 4 for the results of the finite element model presented above also shows that the simulation of several revolutions is not practically feasible using finite elements.

### 3 Elastic Multibody Model

To overcome the large integration times of the finite element model while still having a numerically accurate model, the floating frame of reference approach, see [11], is chosen. Due to the formulation of the deformation with respect to a reference frame that is allowed to undergo large nonlinear motions, the large rotation of the gears can easily be described while still the deformation can be assumed to be linear, thus allowing an efficient modal order reduction. Here, a modally reduced description is used. The equations of motion read

$$\mathbf{M}(\mathbf{z}_I) \dot{\mathbf{z}}_{II} - \mathbf{h}_\omega(\mathbf{z}_I, \mathbf{z}_{II}) - \mathbf{h}_e(\mathbf{z}_I, \mathbf{z}_{II}) - \mathbf{h}_c(\mathbf{z}_I, \mathbf{z}_{II}) = \mathbf{0}, \quad (4)$$

where  $\mathbf{M}$  is the mass matrix,  $\mathbf{h}_\omega$  are generalized inertial forces,  $\mathbf{h}_e$  are generalized internal forces and  $\mathbf{z}_I$  and  $\mathbf{z}_{II}$  are the generalized position and velocity state variables, respectively. All external forces and torques, including contact forces, are given by  $\mathbf{h}_c$ . The generalized position and velocity variables are chosen as

$$\mathbf{z}_I = [\boldsymbol{\rho} \ \boldsymbol{\alpha} \ \mathbf{q}]^T \quad (5)$$

$$\mathbf{z}_{II} = [\mathbf{v} \ \boldsymbol{\omega} \ \dot{\mathbf{q}}]^T \quad \text{with} \quad \dot{\mathbf{z}}_I = \begin{bmatrix} \mathbf{E} & \tilde{\boldsymbol{\rho}} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_R^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{E} \end{bmatrix} \mathbf{z}_{II} = \mathbf{J}^{-1}(\mathbf{z}_I) \mathbf{z}_{II}, \quad (6)$$

where  $\boldsymbol{\rho}$  is the absolute position of the reference frame with respect to the reference frame,  $\boldsymbol{\alpha}$  are appropriate rotation parameters to describe the orientation of the reference frame and  $\mathbf{q}$  are the elastic degrees of freedom. The absolute velocity and the rotational velocity are denoted by  $\mathbf{v}$  and  $\boldsymbol{\omega}$ , respectively. The relationship between the position and velocity states is given by the generalized Jacobian matrix  $\mathbf{J}$  that mainly consists of the rotational Jacobian matrix  $\mathbf{J}_R^{-1}$  that depends on the chosen

rotation parameters  $\alpha$  and the skew symmetric matrix  $\tilde{\rho}$  that introduces the cross product in matrix notation and is derived from the translational positions  $\rho$ , see [9]. Due to the modal formulation, the elastic states  $\mathbf{q}$  are modal coordinates.

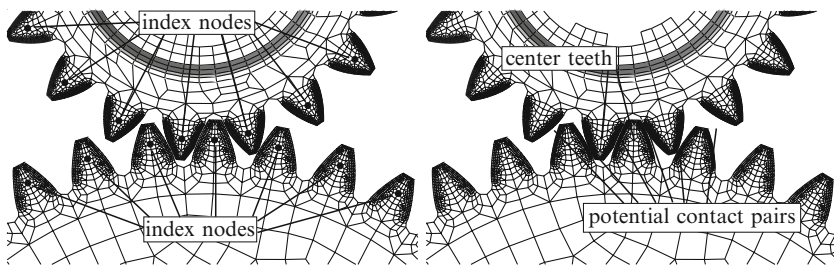
### 3.1 Contact Algorithm

For the contact force calculation a general node-to-surface approach is used. However, if possible, gears are designed with a prime number of cogs, hence the number of possible contact pairs for two gear with  $z_1$  and  $z_2$  number of cogs is  $n_{max} = 2z_1z_2$ . Obviously, a collision detection looping over all flank nodes and elements would be numerically extremely costly, as due to the modal description, a transformation from modal coordinates to nodal coordinates would be necessary for every flank node. Under normal operations typically at most two to three flanks are in contact at the same time. Therefore, a coarse collision detection reducing the node-based contact force calculation to these few flanks is unavoidable for performance reasons.

#### 3.1.1 Coarse Collision Detection

The coarse collision detection is based on index nodes. Each cog is referenced by a separate index node that is approximately located in the center of each cog, see Fig. 6 (left). The absolute positions of the index nodes are calculated in every integration step. Then, the center node, i.e. the index node closest to the center of the associated gear is determined and based on this center node a small number  $z_s$  of cogs on the left and right are considered as possible contact candidates, see Fig. 6 (right). For most designs  $z_s = 2$  is sufficient.

The benefit from this procedure is not only a reduction of possible contact nodes, but also a significant reduction of the size of the transformation matrix needed to transform from modal coordinates to absolute nodal positions. Here, a dynamic reloading scheme is used to dynamically load only the transformation data necessary



**Fig. 6** Coarse contact determination using index nodes (left) and contact candidates from these index nodes (right)

to transform the index nodes as well as the current nodes of the contact candidates. The transformation matrix is only updated if the center node changes.

### 3.1.2 Fine Collision Detection

The fine collision detection is based on a general node-to-surface contact, see Fig. 7. The goal is to find the contact point of the slave node  $Q$  on the surface spun by the nodes  $P_i$  of the surface of the master element.

This is done by finding the point  $\mathbf{x}_c$  with the smallest distance to the slave node  $Q$  which can be formulated by

$$\frac{\partial \mathbf{x}}{\partial \xi}(\xi_c, \eta_c) \cdot [\mathbf{q} - \mathbf{x}(\xi_c, \eta_c)] = 0, \quad (7)$$

$$\frac{\partial \mathbf{x}}{\partial \eta}(\xi_c, \eta_c) \cdot [\mathbf{q} - \mathbf{x}(\xi_c, \eta_c)] = 0. \quad (8)$$

The distance  $g$  between  $Q$  and  $x_c$  is calculated as

$$g = \mathbf{n} \cdot (\mathbf{x}(\xi_c, \eta_c) - \mathbf{q}), \quad \text{with} \quad \mathbf{n} = \frac{\mathbf{x}_{,\xi}(\xi_c, \eta_c) \times \mathbf{x}_{,\eta}(\xi_c, \eta_c)}{\|\mathbf{x}_{,\xi}(\xi_c, \eta_c) \times \mathbf{x}_{,\eta}(\xi_c, \eta_c)\|} \quad (9)$$

and denotes the gap or penetration of the slave node, dependent on the sign of  $g$ , see [12]. For the hexahedral elements used here, the shape function reads

$$\mathbf{x}(\xi, \eta) = \frac{1}{4} \sum_{i=1}^4 (1 + \xi_i \xi) (1 + \eta_i \eta) \mathbf{u}_i, \quad (10)$$

accordingly, (7) and (8) are nonlinear and are solved using Newton–Raphson iteration.

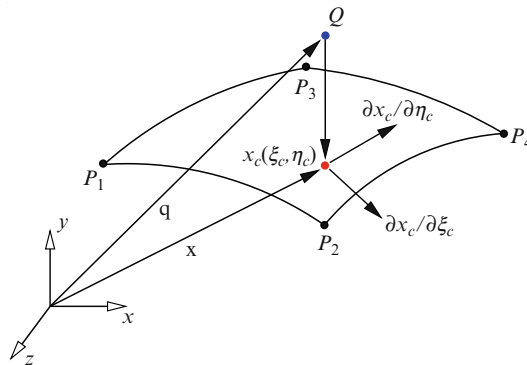
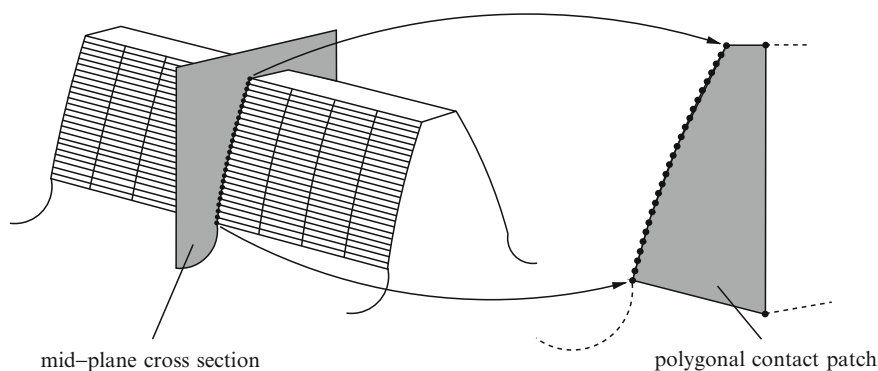


Fig. 7 Fine collision detection as iterative search for contact point



**Fig. 8** Reduction of the three-dimensional contact problem to a two-dimensional polygonal contact problem for perfectly aligned spur gears

When the penetration of the slave node is determined, the nodal contact forces can be calculated using a penalty approach. Then, the nodal contact force for the slave node follows directly from its penetration. The nodal forces acting on the four master nodes representing the master surface follow from the participation factors  $\xi_c$  and  $\eta_c$  of the determined contact point  $x_c$  according to the shape function. When all nodal contact forces are calculated the overall nodal contact force vector can be assembled and transformed to modal coordinates.

For perfectly aligned spur gears the three-dimensional contact situation can be simplified to a two-dimensional contact problem, see [13]. Then, only the nodes in the center plane of the contact candidates are considered and the contact problem is reduced to a polygonal contact problem, see Fig. 8 that can be solved using the ray-crossing algorithm, see [7]. However, in the implementation full three-dimensional contact is used.

After determining which slave nodes have penetrated the polygon defined by the master nodes, the slave nodes are projected on the associated segments. Again, the nodal contact forces for the slave nodes are calculated using a penalty approach and the nodal master forces are calculated according to their participation factors. When all nodal forces are determined for the polygon the contact forces are appropriately distributed to all nodes on the contact line intersecting the respective contact nodes. Since the number of contact nodes is reduced tremendously, this two-dimensional fine collision detection is far more efficient than the three-dimensional cases, however, it may not be as precise as the three-dimensional approach.

### 3.2 Integration

For the calculation of the contact forces many transformations to absolute positions are necessary. Therefore, in order to increase numerical efficiency, the equations of



motion (4) are formulated on acceleration level. With differentiation of equation (6) the equations of motion follow as

$$\begin{aligned} & \mathbf{J}^T \mathbf{M} \mathbf{J} \ddot{\mathbf{z}}_I - \mathbf{J}^T [\mathbf{M} \dot{\mathbf{J}} \dot{\mathbf{z}}_I - \mathbf{h}_\omega(\mathbf{z}_I, \dot{\mathbf{z}}_I) - \mathbf{h}_e(\mathbf{z}_I, \dot{\mathbf{z}}_I) - \mathbf{h}_c(\mathbf{z}_I, \dot{\mathbf{z}}_I)] \\ &= \begin{bmatrix} \mathbf{M}_{tt} & \text{sym.} \\ \mathbf{M}_{rt} & \mathbf{M}_{rr} \\ \mathbf{M}_{et} & \mathbf{M}_{er} & \mathbf{M}_{ee} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\rho}} \\ \ddot{\boldsymbol{\alpha}} \\ \ddot{\mathbf{q}} \end{bmatrix} - \begin{bmatrix} \bar{\mathbf{h}}_t \\ \bar{\mathbf{h}}_r \\ \bar{\mathbf{h}}_e \end{bmatrix} - \begin{bmatrix} \mathbf{h}_{\omega t} \\ \mathbf{h}_{\omega r} \\ \mathbf{h}_{\omega e} \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{h}_e \end{bmatrix} - \begin{bmatrix} \mathbf{h}_{ct} \\ \mathbf{h}_{cr} \\ \mathbf{h}_{ce} \end{bmatrix} = \mathbf{0}. \quad (11) \end{aligned}$$

The second order differential equation (11) is integrated using the central difference scheme, see [2]

$$\ddot{\mathbf{z}}_{I,n} = \frac{1}{\Delta t^2} (\mathbf{z}_{I,n+1} - 2\mathbf{z}_{I,n} + \mathbf{z}_{I,n-1}) \quad (12)$$

and the new displacement  $\mathbf{z}_{I,n+1}$  follows as

$$\mathbf{z}_{I,n+1} = \Delta t^2 \bar{\mathbf{M}}^{-1} (\mathbf{h}_\omega + \mathbf{h}_e + \mathbf{h}_c) + 2\mathbf{z}_{I,n} - \mathbf{z}_{I,n-1}, \quad \text{with} \quad \bar{\mathbf{M}} = \mathbf{J}^T \mathbf{M} \mathbf{J}. \quad (13)$$

The central difference method is a conditionally stable integration scheme and the integration step size has to be smaller than the critical time step  $\Delta t_{crit} = 2/\omega_{max}$ .

In (13) the inverse of the state dependent mass matrix is needed which has to be calculated in every integration step. For a modal model with properly scaled eigenmodes, the sub-matrix  $\mathbf{M}_{ee}$  in (11) is an identity matrix. This allows to use the structure of the mass matrix for an efficient calculation of the inverse

$$\bar{\mathbf{M}}^{-1} = \begin{bmatrix} \tilde{\mathbf{M}} & \text{sym.} \\ [\mathbf{M}_{et} \ \mathbf{M}_{er}] \tilde{\mathbf{M}} \ \mathbf{I} + [\mathbf{M}_{et} \ \mathbf{M}_{er}] \tilde{\mathbf{M}} [\mathbf{M}_{et} \ \mathbf{M}_{er}]^T \end{bmatrix} \quad (14)$$

with

$$\tilde{\mathbf{M}} = \left( \begin{bmatrix} \mathbf{M}_{tt} & \mathbf{M}_{rt}^T \\ \mathbf{M}_{rt} & \mathbf{M}_{rr} \end{bmatrix} - [\mathbf{M}_{et} \ \mathbf{M}_{er}]^T [\mathbf{M}_{et} \ \mathbf{M}_{er}] \right)^{-1}. \quad (15)$$

Now the inverse of the mass matrix can be calculated by only inverting the  $6 \times 6$  matrix and a few matrix multiplications. This is not only much faster, but also much more precise than an iterative inversion of the full mass matrix.

Constraints, like the ideal rotational joints in Sect. 2 can be incorporated by adding constraint equations to (11), eventually resulting in a differential-algebraic equation. However, this would destroy many of the measures taken to improve performance. As all constraints used with the models from Sect. 2 are scleronomic, they can easily be integrated into the equations of motion for comparison with these models. In more complicated cases, a penalty approach is used to model constraints.

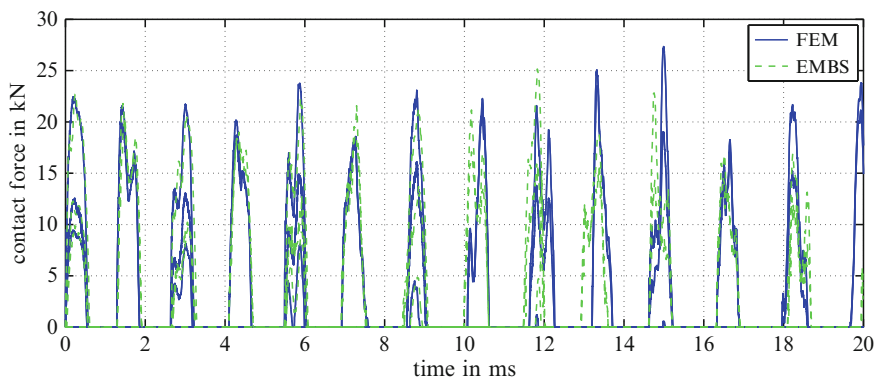
## 4 Simulation Results

In the following, simulation results for two gear pairs are given. The first gear pair is the pair of spur gears presented in Sect. 2. The simulation results from the elastic multibody model are compared to the finite element solution and the numerical efficiency is evaluated on the basis of these simulations. The second gear pair consists of two helical gears with cylindrical gear bodies. Besides contact forces and rotational velocities, joint forces are investigated and the results are compared to a spur gear pair of the same size.

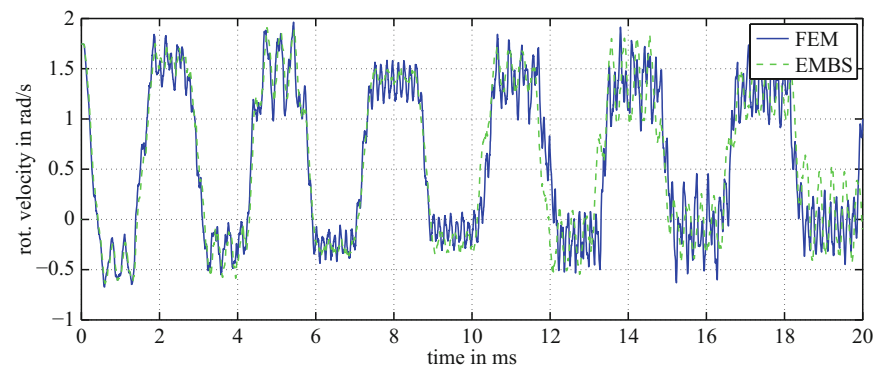
### 4.1 Spur Gears

Figures 9 and 10 show a comparison between the results of the finite element model and the results obtained from the modally reduced elastic multibody model for the spur gear pair. For the simulations shown here about 1,500 modes per gear were used. This large number at first seems unnecessary, but as shown in [14], modes up to an eigenfrequency of about 80 kHz are necessary to get correct contact forces. This is mainly due to the local impulsive contact with very local deformations in a small contact area, see [10].

Both contact forces and rotational velocities agree very well to the finite element solution. For the first few contacts, the contact forces almost perfectly match the finite element solution in shape, magnitude and particularly the number of cogs simultaneously in contact. The rotational velocities, too, agree very well and show the torsional vibration of the outer rim against the hub during the free flight phase. The elastic multibody model is able to describe the elastic effects efficiently and very precise results can be obtained.



**Fig. 9** Contact forces for the elastic multibody model and the finite element model



**Fig. 10** Rotational velocities for the elastic multibody model and the finite element model

**Table 1** Requirements for integration time, disk space and memory for the spur gear pair and 13 contacts using three different models

Model	Integration time	Disk requirement	Memory requirement
MBS	5 s	$\ll 0.1$ GByte	$\ll 0.1$ GByte
FEM	6,98,400 s	$\approx 1$ GByte	$\approx 1$ GByte
EMBS, 2d-contact	683 s	$\approx 8$ GByte	$\approx 0.5$ GByte
EMBS, 3d contact	1,834 s	$\approx 8$ GByte	$\approx 2$ GByte

4.2 Numerical Efficiency

The results obtained from the elastic multibody model are almost as precise as the reference finite element solution. However, the motivation for the elastic multibody model is to get a numerically much more efficient description of gear contact. To evaluate the numerical efficiency, the integration time and several hardware requirements are summarized in Table 1. The computation times were measured on a Pentium 4 computer with 2 Gigabytes of RAM.

Regarding the integration time and the time for setting up the model setup, the rigid body model is fastest. Though, as shown, the results are completely wrong for very elastic gears. The finite element model considers all elastic effects and, due to the fine discretization, can be regarded to give very precise results. On the other hand, the integration time is enormous. The elastic multibody model needs pre-processing. This pre-processing includes geometrical analyzes to extract contact nodes and surfaces from the finite element mesh and, most expensive, a modal analysis of each gear. However, the modal analysis only has to be carried out once per gear and is stored to disk, explaining the large disk requirements. In all subsequent simulations, the pre-processed data can be used which shows in the integration

**Table 2** Time distribution during one integration step (*left*) and one step of the 3d-contact calculation (*right*) for the elastic multibody model

Task	Percentage	Task	Percentage
Contact calculation	77%	Transformation of contact nodes	41%
Integration	19%	Fine collision test	27%
Miscellaneous	4%	Coarse collision test	25%
		Miscellaneous	4%
		Transformation of nodal forces	3%

time, being about a factor of about 1,000 faster than the finite element analysis when the 2d-contact for spur gears is used and still about 400 times faster using the 3d-contact.

To reveal the most costly parts in an integration step for the elastic multibody system, Table 2 (*left*) shows the percentage fraction of the major parts of one integration step. For reasonably discretized flank surfaces, the most costly parts of one integration step are the integration itself and the contact calculation. The times necessary for one contact calculation step are broken down in Table 2 (*right*) revealing three big blocks consuming almost 90% of the total time altogether.

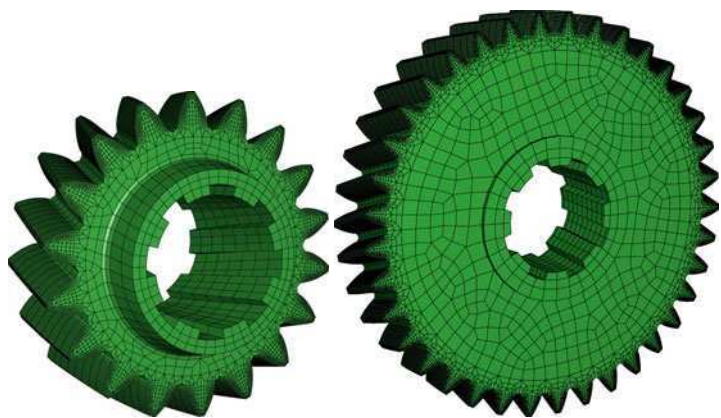
The coarse collision detection accounts for about a quarter of the time needed for one contact calculation step. However, since the values represent average elapsed times evaluated from the considered example, these 25% partially include the time needed to load the transformation data from disk as well. The by far most costly part is the transformation of the contact nodes that strongly depends on the discretization. Even though the model is fully implemented in Matlab, here, a further optimization is hardly possible, since the transformation matrices are already dense and Matlab uses a highly optimized basic linear algebra kernel. The fine collision detection accounts for about 27% of a contact step. Here, the Matlab implementation is optimized by implementing the fine collision detection in C.

### 4.3 Helical Gears

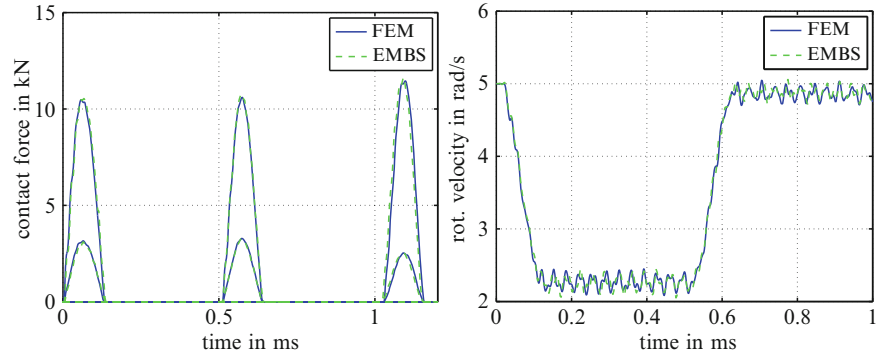
The second investigated gear pair is a helical gear pair with helix angle  $\beta = 15^\circ$ ,  $z_1 = 18$ , and  $z_2 = 37$ , see Fig. 11.

Again, the two gears are mounted with ideal rotational joints and both gears are given an initial rotational velocity of  $\omega_{10} = \omega_{20} = 5 \text{ rad/s}$ . However, this time the rotational joints for the elastic multibody model are modeled using a penalty approach. This easily allows to calculate hub forces without explicitly having to solve constraint equations.

Figure 12 (*left*) shows the contact forces for three contacts for the finite element and the elastic multibody model. Once more, the calculated contact forces agree very well.



**Fig. 11** Helical gear pair with cylindrical gear bodies

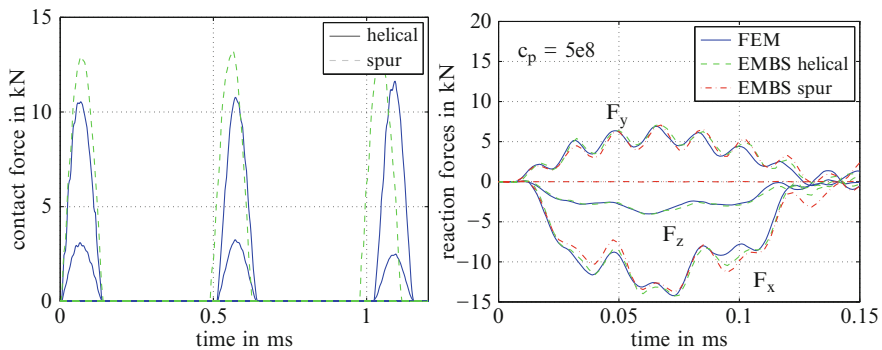


**Fig. 12** Contact forces for the elastic multibody model and the finite element model for helical gear pair

The rotational velocities presented in Fig. 12 (right), too, show a very good agreement during all three contacts. However, the dynamical effects discussed above are not as apparent as for the spur gear pair for two reasons. First, the gear body of both gears are almost cylindrical and, therefore, very rigid considerably reducing elastic effects.

Second, the helix angle results in an increased overlap ratio which in turn leads to two flanks in simultaneous contact for all three contacts. In contrast, as shown in Fig. 12 (left), a similar spur gear pair with helix angle  $\beta = 0^\circ$  only shows one flank in contact at a time. Therefore, the contact forces are much higher and the contact is shorter and harder for the spur gear.

Another effect of the helix angle is that axial reaction forces are introduced, as shown in Fig. 13 (right) for the pinion during the first contact. Clearly, for elastic multibody models of the helical gear pair an axial reaction force can be observed that agrees very well to the finite element model again, whereas the spur gear pair



**Fig. 13** Comparison of joint forces between the finite element model and the elastic multibody system and spur and helical gear

only shows radial components. More surprisingly, however, the reaction forces of the spur gear pair and the helical gear pair show strong vibrations even though the contact forces as well as the rotational velocities show almost no vibrations. That shows that even for fairly rigid gears an elastic approach is necessary to compute precise results.

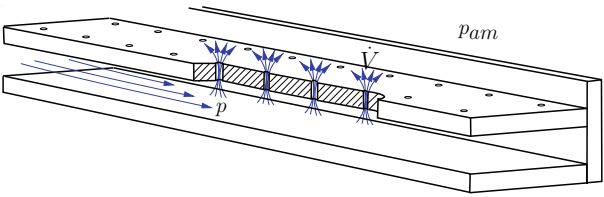
## 5 Experimental Results

To validate the proposed elastic multibody model, basic experimental impact investigations have been carried out. For these investigations, a simple impact body with cuboid shape is impacting on a spur gear. The gear used for the experimental investigations has  $z = 37$  teeth of modulus  $m = 4.9\text{ mm}$  and a cylindrical gear body. To reduce the experimental complexity, we limit ourselves to stationary gear wheels, which tremendously reduces the measurement setup, particularly the alignment of impact body and gear wheel.

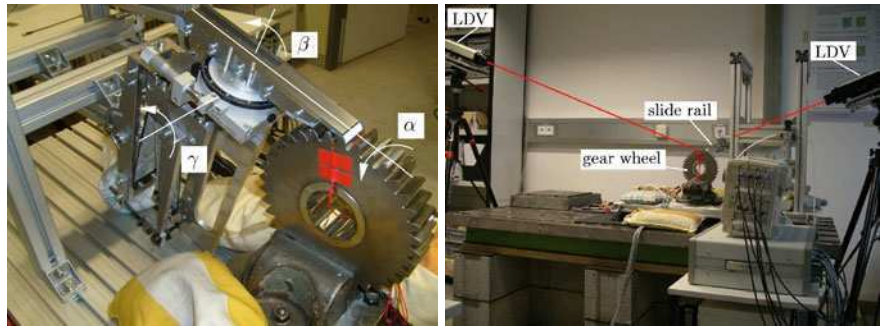
To guarantee a very precise guidance of the impact cuboid, a guiding slide rail is used. To allow reproducible experimental conditions and negligible friction, the slide rail is supplied by pressured air that generates an air cushion upon which the cuboid hovers almost frictionless, see Fig. 14.

Furthermore, some air exhausts from the gap between cuboid and slide rail, producing a self-aligning effect. The slide rail is mounted on a precision rotary stage that allows to adjust the alignment angle  $\beta$  with a precision of less than  $1/1000^\circ$ . The rotary stage itself is mounted on a frame that allows to adjust the alignment angles  $\alpha$  and  $\gamma$ , see Fig. 15 (left). The complete experimental setup is shown in Fig. 15 (right).

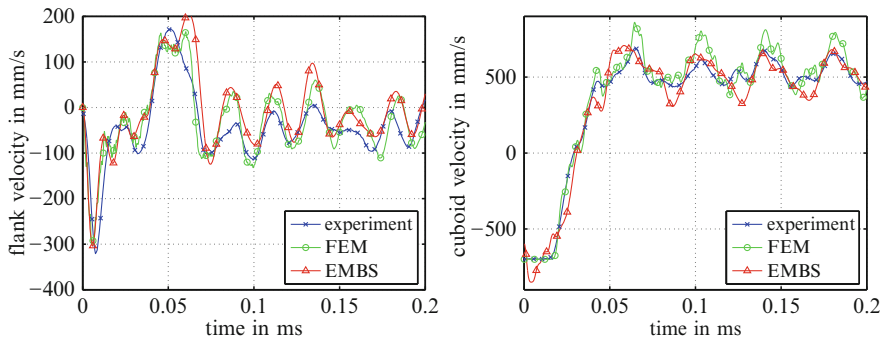
Generally, the direct measurement of contact forces is difficult. Therefore, flank velocities, as well as strains are measured in close vicinity to the contact area and compared to corresponding simulation results. Laser-Doppler vibrometers are used



**Fig. 14** Schematic drawing of the air supplied slide rail



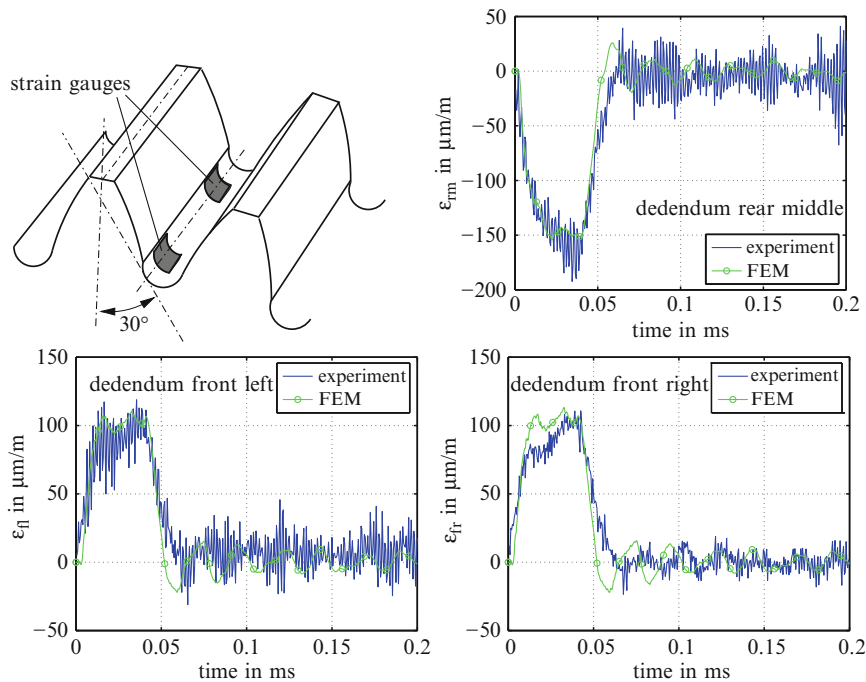
**Fig. 15** Rotary stage and supporting frame for cuboid impact body (*left*) and overall experimental setup (*right*)



**Fig. 16** Flank velocities in the middle of the rear flank (*left*) and cuboid velocity (*right*) for experiment, finite element model and elastic multibody model

to measure the flank velocity on the back side of the impacted tooth and strain gauges are applied in the dedendum. Since the interesting frequencies are up to 80 kHz, DC amplifiers instead of carrier frequency amplifiers are used, to directly measure the mistune of the bridge.

Figure 16 shows the flank velocity measured in the middle of the rear flank and the cuboid velocity for one experiment together with the simulation results from the finite element model and the elastic multibody model. Apparently, the results agree very well.



**Fig. 17** Position of the strain gauges in the dedendum and strain signals for one impact for experiment and finite element model

The measured and simulated strains in the dedendum are shown in Fig. 17. The signal of the strain gauges turns out to be very noisy which is mainly due to the use of DC amplifiers. To reduce the noise ten measurements have been averaged. Here too, a good agreement between simulation and experiment can be observed.

Comparing the experimental results with simulations, apparently, the finite element model and the elastic multibody model give very good results. Both the surface velocities on different teeth, as well as strains in the dedendum can be calculated precisely and the good agreement between simulation and experiment even holds for a long time after the contact. For the basic impact conditions described above, the elastic multibody model can be regarded as validated.

## 6 Conclusions

A very detailed finite element model and a classical rigid body model have been used to investigate several contacts between a gear pair taken from a real technical application. It was shown that the elasticity of the gear body has a significant influence on the contact forces as well as on the global motion and that simple rigid body models cannot be used to calculate precise contact forces. Instead, a fully elastic approach is necessary.



To avoid the tremendous numerical effort needed to solve the full finite element model, a modally reduced elastic multibody model with contact has been presented. However, several implementation issues must be considered carefully to get a numerically efficient implementation and to reduce the memory requirements needed for contact node transformations. Particularly a coarse collision detection including a dynamic reloading scheme is important. Besides a general tree-dimensional node-to-segment approach, a very efficient contact algorithm for perfectly aligned spur gears is used. In combination with an explicit integration scheme which exploits the structure of the mass matrix for its inversion, this elastic multibody model allows very precise simulations in short time compared to finite element solutions. The presented approach is general enough to also simulate helical gears and simulation results for a helical gear pair were shown and compared to results of a similar spur gear pair.

In order to validate the numerical models, basic experimental impact investigations have been carried out, where a stationary gear wheel was impacted with simple impact bodies and flank velocities as well as strains in the dedendum have been measured. Comparing simulation and experiment, a very good agreement was found.

**Acknowledgements** Most of the work presented here is related to the projects “Rädertriebsimulation” and “Rädertriebsimulation II” which are supported by the Forschungsvereinigung Verbrennungskraftmaschinen (FVV) and the Forschungsvereinigung Antriebstechnik (FVA). This support is highly appreciated and we would like to thank all industrial members of the “Arbeitskreis Rädertriebsimulation” for their contributions and ideas.

## References

1. Bathe KJ (1996) Finite element procedures. Prentice-Hall, Englewood Cliffs
2. Cook RD, Malkus DS, Plesha ME, Witt RJ (2002) Concepts and applications of finite elements analysis. 4th edn, Wiley, New York
3. DIN Deutsches Institut für Normung, DIN 3990-1 (1987) Tragfähigkeitsberechnung von Stirnrädern – Teil 1: Einführung und allgemeine Einflußfaktoren (in German). Beuth Verlag, Berlin, Germany
4. International Organization for Standardization, ISO 6336-1 (2000) Calculation of load capacity of spur and helical gears – Part 1: Basic principles, introduction and general influence factors
5. Kahraman A, Blankenship GW (1997) Experiments on nonlinear dynamic behaviour of an oscillator with clearance and periodically time-varying parameters. *J Appl Mech* 64:217–226
6. Niemann G, Winter H (1989) Maschinenelemente – Band 2 (in German). Springer, Berlin
7. O'Rourke J (1993) Computational seometry in C. Cambridge University Press, Cambridge
8. Padieth R (1978) Im Abwälzverfahren hergestellte Verzahnungen: Exakte Ermittlung der Zahnform (in German). *Antriebstechnik* 17(10):434–436
9. Schwertassek R, Wallrapp O (1999) Dynamik flexibler Mehrkörpersysteme (in German). Vieweg, Braunschweig
10. Seifried R (2005) Numerische und Experimentelle Stoßanalyse in Mehrkörpersystemen. Dissertation, Schriften aus dem Institut für Technische und Numerische Mechanik der Universität Stuttgart, Band 2. Shaker Verlag, Aachen
11. Shabana AA (2005) Dynamics of multibody systems. Cambridge University Press, Cambridge

12. Wang SP, Nakamachi E (1996) The inside-outside contact search algorithm for finite element analysis. *Int J Numer Meth Eng* 40:3665–3685
13. Ziegler P, Eberhard P (2008) Simulative and experimental investigation of impacts on gear wheels. *Comput Meth Appl Mech Eng* 197(51–52):4653–4662
14. Ziegler P, Eberhard P, Schweizer B (2006) Simulation of impacts in geartrains using different approaches. *Arch Appl Mech (Ingenieur Archiv)* 76(9):537–548